August 2021

Applications of Set-Theoretic Topology in the Construction and Analysis of Engineering Design Spaces

Joshua Bronson Ortiz
Clemson University, joshortizbilling@gmail.com

Applications of Set-Theoretic Topology in the Construction and Analysis of Engineering Design Spaces

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mechanical Engineering

---

by
Joshua Ortiz
August 2021

Accepted by:
Dr. Joshua Summers, Committee Chair
Dr. James Coykendall
Dr. Cameron Turner

Abstract

The idea of design spaces in engineering has appeared in many forms and served a variety of purposes in practice, research, and literature. Yet very few of the definitions put forth have a concrete mathematical structure that can be practically applied by the designer in real-time. This research seeks to address this gap by taking advantage of tools and techniques in point-set topology, a field that has been used successfully in a number of different areas. The primary objective of this undertaking is to formalize definitions for design spaces as topological structures that will encapsulate many of the relevant characteristics of both the problem to be solved and the designs that are being considered.

Three separate spaces are presented: the problem space, the solution space, and the quality space. The problem space is defined by the requirements that pertain to the problem and represents the target that designs must hit to be considered a solution. The solution space is the collection of design embodiments for a given concept that meet the specified constraints. Finally, the objective space is a space that allows different design concepts to be compared to one another based on common criteria that any solution would exhibit. Along with these definitions, several methods are also proposed to operate on the design spaces to assist in their analysis and comparison. Measures are introduced for assessing the similarity of spaces as they evolve and for quantifying how sensitive solution spaces are to changes in requirements. Also, a process for gauging the relative utility of different concepts is presented. Two examples are included to demonstrate implementation, one simplistic for explanatory value and the second more complex to show scalability.

Topology has been demonstrated to be a versatile and extensible lens for data interpretation and exploration. Given this adaptability, it is hoped that this thesis will serve as a foundation upon which future work can build so that a wide array of novel capabilities can be established for engineers, designers, researchers to draw upon in their pursuits.

Table of Contents

List of Tables

List of Figures

List of Symbols

Alphabetic

$\mathcal{A}$: the set of constraint parameters associated with a design project

$\mathcal{B}$: the set of design variables associated with a design project

$\mathcal{C}$: the constraint space, embedding for the problem space

$\mathcal{F}$: the form space, embedding for the solution space

f: a generic function

H: a candidate design concept

J: Jaccard index

${}^{p}\mathcal{M}$: the problem map associated with a design concept

${}^{q}\mathcal{M}$: the quality map associated with a design concept

m: dimensionality of the form space (the number of design variables) –or– symbol for mass

n: dimensionality of the constraint space (the number of constraint parameters)

$\mathcal{O}$: the objective space, embedding for the quality space

oc: overlap coefficient

$\mathcal{P}$: the problem space

$\mathcal{Q}$: the quality space

$\mathbb{R}$: the set of real numbers

$\mathcal{R}$: the set of requirements associated with a design project

$\mathcal{S}$: the solution space

ss: spatial similarity index

U: the utility response surface associated with a design project

u: utility curve for a specified criterion

w: relative importance (weight) for a specified criterion

X: a generic set or space

Y: a generic set or space

x: a generic value or point in space

y: a generic value or point in space

Greek

α: a single constraint parameter, element of $\mathcal{A}$

β: a single design variable, element of $\mathcal{B}$

Γ: the set of criteria associated with a design project

γ: a single criterion, element of Γ

μ: a generic measure (e.g. volume, cardinality)

τ: a generic topology

Ω_X: the sample space of a set X

Accents and symbols

Breve ($\breve{X}$): indicates a set of points sampled from Ω_X

Tilde ($\tilde{J}$): indicates a modified form the function or equation J

Delta (ΔX): indicates a change to a set or space X

Del ($\bigtriangledown X$): indicates a change gradient over the set or space X

Scripts

Prescript (${}^{q}\mathcal{M}$): denotes a specific nature/usage (e.g. quality map versus problem map)

Superscript (H^A): denotes a relation to a distinct design concept

Subscript ($\mathcal{P}_2$, $\mathcal{R}_\alpha$): denotes an index value over a set –or– an association with another object

Chapter 1

Introduction

1.1 Prior Work

The word “space” has been used in a number of different ways in design research, often paired with a more specific name such as ”design space”, ”problem space”, or ”solution space”, among others. Despite its ubiquity, the meaning of the term in any given context is often only implied and left to the reader to ascertain from the usage.

Some research, however, has sought to provide formal definitions for this terminology that offer more precise language for discussion, though individual objectives have varied. Goel and Pirolli (1992), for example, described the design problem space as a formalized structure of the problem-solving process characterized by the people involved and the context of the task. They hypothesized that the structures of these spaces are invariant across all design scenarios, but that they differ in nature from problem-solving structures in non-design scenarios. The formalism they proposed therefore defines the spaces according to the characteristics they exhibit, implying that the space itself is an abstract notion that exists apart from its definition. The authors also offered no prescription for how the spaces can or should be used by designers and did not consider the solution or group of possible solutions as a space unto itself. Their stated goal, rather, was to use the notion of the design problem space as an explanatory tool for understanding the nature of the design process.

Bowen and Dittmar (2017) offered a more rigorous definition for design spaces. Their model uses a set-based approach that groups together all aspects of a design. Similar to Goel

and Pirolli (1992), they include the designer, the process, and the environment as part of a single cohesive construct. Their aim in doing so, however, was to provide designers with a formal means of establishing and tracking the relationships between various elements as they evolve during the design process. As such, their definitions are mathematical in nature and are presented with formal logic for the combination and progression of the space along the way. To summarize their formalism, a design space is a snapshot of the design process during a short segment in time. The basic unit is a 'simple' design space (Figure 1.1) defined as a 5-tuple $DS = (DT, Us, D_{entry}, D_{exit}, DRel)$ representing the design team DT, the various user groups $Us = U_1, U_2, ...$, the design artifacts received from stakeholders as the segment begins D_{entry}, the artifacts provided to stakeholders at the end of the segment D_{exit}, and the relationships between artifacts $DRel$. More complex spaces can then be built by nesting simple spaces as needed to capture the nature of real-world design projects. With these two types of spaces, so-called traces could then be formulated by creating an ordered sequence of design spaces – either simple or complex – and associating each with a unique point in time. This allows each step of the process to be tracked and the evolution to be examined. The authors readily admit that their model views the process from a high level, which may give the impression of over-simplification. In practice, any given artifact or relation of a space would, in general, be a highly detailed and complex construct unto itself. The primary contribution of their paper, then, is to provide a means for recording the progress of design in an organized manner which can be applied to highly sophisticated design efforts with divided teams, multiple stakeholders, and tiered subsystems.

The efforts of both Goel and Pirolli (1992) and Bowen and Dittmar (2017) provide broad and somewhat abstract interpretations for the idea of spaces within the context of design. Yet neither provides a specific tool that a designer might use to improve upon a given design. Goel and Pirolli (1992) focus on the human aspect of design research while Bowen and Dittmar (2017) were concerned with tracking changes. Both are important from a certain point of view, but not geared for practical use in arriving at a better solution sooner. For that, one must provide a mechanism by which to analyze or evolve the space in an objective and meaningful way.

Maher et al. (1996) provided the framework for such a mechanism through the process of co-evolution. The authors claimed that many tools treat design as a well-defined problem, and therefore become cumbersome in the face of change. Their formulation sought to capture the inter-reliance between the objectives and the solution of a design effort and also offered a process for

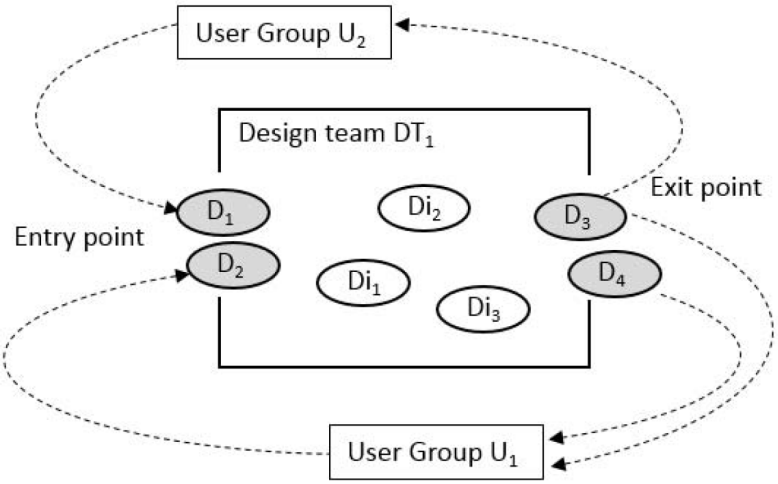

Figure 1.1: An example of a 'simple' design space from Bowen and Dittmar (2017).

exploring possible characterizations of them. It revolves around two distinct but intertwined design spaces, a problem space, and a solution space, both of which are dependent on time. Figure 1.2 shows an example of the interconnection between the spaces over time. The problem space $P(t)$ is described as the design goal at time t, and the solution space $S(t)$ is the corresponding search space for a solution that meets that goal. The two spaces interact via fitness functions. The fitness of a solution s_i in $S(t)$ at any given timestep is defined by how well it satisfies the current collection of criteria that make up $P(t)$. Likewise, the fitness of a criterion p_j in $P(t)$ is determined by the number of designs in $S(t)$ which meet that criteria. The proposed evolutionary mechanism for the two spaces is a genetic algorithm wherein the solutions and criteria are encoded as binary strings and then propagated, combined, mutated, or killed based on their fitness to create the next generation of spaces for timestep $t+1$. In this way, the spaces change together with each influencing the other. Throughout this cycle, the designer retains the ability to add desirable design solutions and criteria at each iteration or to remove unacceptable ones.

Another mechanism for design space evolution involving topological structures was proposed by Braha and Reich (2003). While the authors state at the outset that their process would be cumbersome to use in practice, it nevertheless provides an example of how topology can be used to describe abstract notions about the way design happens and how to improve the methods used. Their representation focuses on functional properties f that designate "the behavior that an artifact

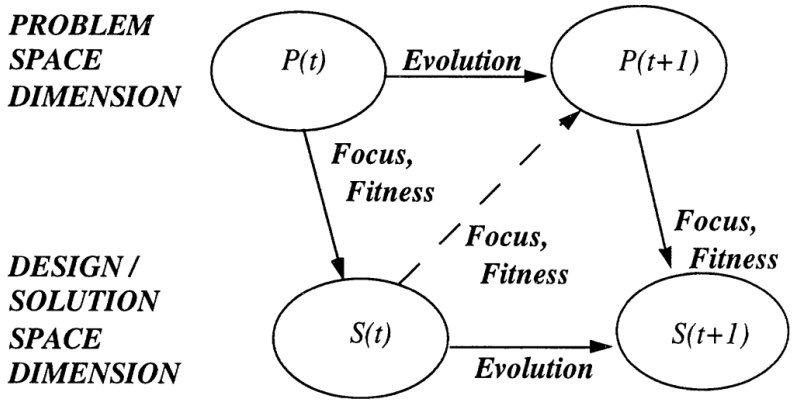

Figure 1.2: The problem-design exploration model from Maher et al. (1996).

displays when it is subjected to a situation" as well as structural descriptions d that delineate the "observable or otherwise measurable attributes" of an individual design. The function space F, then, is the set of all possible functional descriptions of the artifact that would provide the design with the necessary behavior. Similarly, the structure space D is the set of all structural descriptions of the design that could be generated with the designers' knowledge at any given time. An all-encompassing design space arises from the combination $F \times D$, which gives a set of function-structure pairs $\langle f_i, d_i \rangle$. The authors go on to outline the progression from one functional (respectively structural) description to the next $f_i \rightarrow f_{i+1}$ (respectively $d_i \rightarrow d_{i+1}$) in a process called refinement. To do this, a closure space (all possible reachable descriptions) of the current description is determined, from which the best available description is chosen as next in the sequence. This new iteration will have an overlapping but presumably somewhat different closure space as compared to its predecessor, and, in turn, will produce a new refinement. This process is summarized in Figure 1.3. The existence of two mapping functions are also postulated that allow for transition from F to D and vise versa, referred to as synthesis and analysis, respectively. The authors intentionally, and perhaps necessarily, leave the realization of these maps and the specifics of how to define "reachable states" and "best description" up to the designer to determine according to the details, capabilities, and priorities of their project.

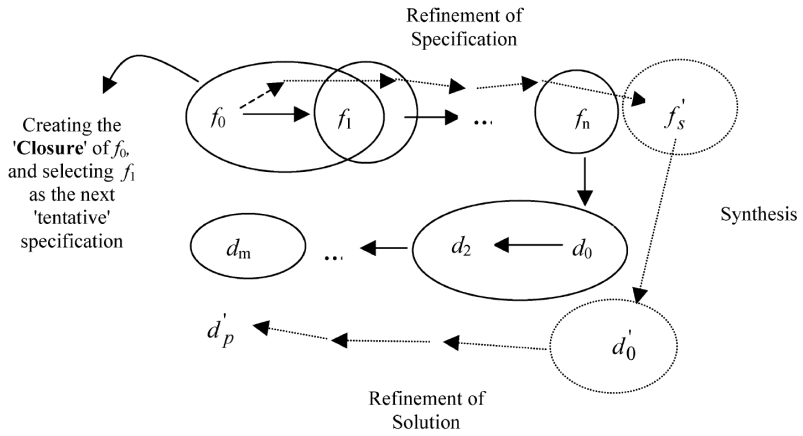

Figure 1.3: Design process model from Braha and Reich (2003) with parallel candidates being developed. Each oval denotes the closure of a given description (functional or structural).

1.2 Motivation

As with many other mathematical formalisms in engineering design, the work of Maher et al. (1996) and Braha and Reich (2003) constitute high-level abstractions with little practical application as stand-alone tools. The benefits to research notwithstanding, designers, too, may yet profit from these frameworks if the details of precisely how to perform the necessary mappings and representations can be filled in. That is where the definitions and methods discussed in this paper make a contribution. In Section 2.1.1, a formalism is presented which describes in detail how a text-based requirement can be represented as a topological entity upon which calculations can be performed. The spaces themselves will be outlined in Sections 2.1.2 and 2.1.4,with the mapping between them described in Section 2.1.3. Then, throughout the remainder, these will be demonstrated and examined with a focus on both closed-form mathematical constructs as well as statistical tools such as surrogate models and computational aides like ANSYS. The end goal is to bridge the gap between the theoretical concepts summarized above and the tools used every day by designers and engineers.

Here, as in Braha and Reich (2003) and Maher et al. (1996), the topological notion of spaces will be used here as the mathematical foundation for the proposed formalism, but with an emphasis on practical implementation. As has been demonstrated in other fields and in other ways (Ruiz-Pérez et al., 2016; Zhu and Gao, 2016; Wasserman, 2018; Siddique and Rosen, 2001; Taura and Yoshikawa, 1994), topology can provide powerful tools for use both in construction and analysis of these spaces.

With careful interpretation, it can help to shed light on the design concepts to which they ascribe. Moreover, the transfiguration of design abstractions into specific topological constructs allows for the development of ever more sophisticated techniques that can be used in both design processes and in research settings.

Given the state of current research and with the aim of creating a tool for real-time use in design, the overarching objectives of this research are to develop a formal structure for design spaces and their interactions that:

1. Has an explicit mathematical defintion
2. Has practical applications to engineering design scenarios
3. Can be extended and improved upon by others in the future
4. Will address gaps identified in current literature

The remainder of this document focuses on introducing, explaining, and demonstrating a model that satisfies these goals.

1.3 Background

In engineering, the term *topology* usually has certain connotations. Often it refers to the physical shape of an object, as in *topological optimization*. However, in this thesis, the word is used in its stricter mathematical sense. Specifically, topology in this context is intended to mean *point-set topology*, also known as *general topology*. Wolfram MathWorld describes point-set topology as "the ground-level of inquiry into the geometrical properties of spaces and continuous functions between them" and that it constitutes the "foundation on which the remainder of topology (algebraic, differential, and low-dimensional) stands" (Weisstein).

As will become clear in subsequent sections, most engineers will already be familiar with many of the implementations of topology used throughout this formulation. However, in the interest of mathematical rigor and clear communication, it is worth defining the necessary terms and constructs in the context of point-set topology to ensure a solid foundation both for the remainder of this paper and to enable consistency for future development of the proposed models and methods.

Many concepts in topology are defined in terms of sets and their operations. A *set* is simply a collection of unique, unordered items. Any item x contained within a set X is referred to as a *member* or *element* of that set (denoted $x \in X$). These elements are often numeric, but that is not strictly necessary, they can be anything. Indeed, some of the sets discussed later will be composed of items that are explicitly non-numeric. Furthermore, a set may be empty, denoted by $\varnothing$, or it may contain an infinite quantity of items. The number of elements of a set is called its *cardinality* ($|X|$) and a set that is wholly contained within another set is a *subset* ($\{1,2\} \subset \{1,2,3,4,5\}$). Note that any set may be considered a subset of itself ($X \subseteq X$), but not a *proper subset* ($X \not\subset X$).

The set operations most often used here are the *union* and *intersection*. For two sets A and B, the *union* ($A \cup B$) is the set that contains all elements of A and all elements of B; in other words, those elements that are found in either A or B. The *intersection* ($A \cap B$) is the set that contains only those elements which are found in both A and B. To illustrate, let $A = \{x, 1, 9\}$ and $B = \{x, y, 9\}$. Then $A \cup B = \{x, 1, y, 9\}$ and $A \cap B = \{9, x\}$. There is one other set operation that is used occasionally in this paper, which is the Cartesian product. This operation generates a new set comprised of all possible pairs of the two original sets as ordered pairs; each pair is considered a single member of the new set. For A and B defined previously, $A \times B = \{(x,x), (x,y), (x,9), (1,x), (1,y), (1,9), (9,x), (9,y), (9,9)\}$. The cardinality of the Cartesian product is the scalar product of the cardinality of the two initial sets. In this case $|A| = 3$ and $|B| = 3$, so $|A \times B| = 9$.

The basic entity of point-set topology is the *topological space*. A *topological space* is a pair of sets (X, τ), where X is a set of elements and τ is a collection of subsets of X, satisfying the following three conditions:

1. The empty set $\varnothing$ and X itself must both be members of τ
2. Any arbitrary union of elements of τ is also an element of τ
3. The intersection of any finite number of members of τ is in τ

The set τ is called *a topology* on the set X and it can be loosely thought of as the collection of all available relationships between the members of X. A specific topology for a set may be defined explicitly by an exhaustive list containing each set of related members of X, or implicitly by a rule that describes how to determine the relation given a subset of members. Often, when the intention

is clear, the space (X, τ) is referred to only as X with the existence of τ being implied. Though it should be noted that a set X may have more than one possible topology and each will result in the space having different topological characteristics.

The most important way that different topologies come into play for this paper is in the case of *metric spaces*. A *metric space* is a topological space whose topology arises from a *metric*, or distance function. Such a function is an example of a rule that implicitly defines a topology, as mentioned earlier. One of the most common metrics is the "Euclidean metric" (L^2 distance) and the space it gives rise to is called "Euclidean space". This is probably the type of space most commonly encountered by engineers in their training. Other common metrics include the "taxicab metric", or L^1 distance, and the "discrete metric". More generally, given a space X and points $x, y, z \in X$, a metric is any function that takes a pair of points in the space X and outputs a non-negative real number ($d : X \times X \to \mathbb{R}^+$) such that:

1. The distance from x to y is 0 if and only if x and y are colocated. $d(x, y) = 0$ iff $x = y$
2. The distance from x to y is the same as from y and x. $d(x, y) = d(y, x)$
3. The distance from x to z is less than or equal to the sum of the distances from x to y and from y to z. $d(x, z) \leq d(x, y) + d(y, z)$

Another important concept for this work is that of a *measure*. A measure is a function that assigns a non-negative real number (or positive infinity) to a given set. This number can be most intuitively thought of as something of a 'size' for the set in question. As with metrics, there are different measures on sets. Cardinality (also called *counting measure*), mentioned above, is perhaps the most straight-forward interpretation of this for a finite set. However, another type of measure that engineers will be familiar with is *volume*. Though often viewed as the physical space taken up by an object in three dimensions, the idea is mathematically extensible to any number of dimensions. The more general term for volume is the *Lebesgue measure*. In dealing real numbers, the Lebesgue measure is the generalization of the concepts of length, area, and volume in 1, 2, and 3 dimensions, respectively, but extends the idea to any number of higher dimensions as well. For higher dimensions, it is sometimes referred to as n-dimensional volume or hypervolume. There are many other types of measure as well, but cardinality and the Lebesgue measure are the two most important ones for the purposes of this formalism, used often for the definitions in Chapter 2.

The topics discussed above serve as the foundational mathematical concepts upon which the formalism proposed in this paper is based. While not a full overview of the realm of topology, this groundwork will allow for future research to expand on the formalism by taking advantage of deeper and more complex topological structures and techniques. Moreover, any existing tools which rely on these underlying concepts may be adapted for use with the methods discussed here.

Chapter 2

Topological Design Spaces

2.1 Proposed Formalism

This research posits that there are three fundamental types of space that need to be considered in engineering design: a space that bounds the design goals, the problem space; a space that contains solutions that meet those goals, the solution space; and a space in which dissimilar design concepts may be compared, the quality space. The first two are similar to the spaces discussed previously in the context of other work, but the latter is necessitated by the manner in which solution spaces are defined here. Solutions may come in different forms to solve the same problem – consider the case of electric versus internal combustion vehicles for commuting to work. For that reason, solution spaces for dissimilar design concepts may not provide a mathematical means for direct comparison. The quality space addresses that by providing common characteristics or performance metrics against which all concepts that address a specific problem can be contrasted and evaluated.

It is relatively common in engineering design to use design constraints as a sort of boundary which separates solutions from non-solutions. Although constraints are often adjusted during the design process, for their duration each one serves as a definitive line between satisfaction and dissatisfaction. For this reason, one may say that the problem itself is defined according to where these lines are drawn. Constraints do not communicate what the solution looks like, but they do determine whether or not a design can be considered a solution. A commuter vehicle must be road-legal, for example, in order for it to serve its intended purpose. Anything not road-worthy would not be a solution to such a design problem. The proposed formalism acknowledges this fact and makes literal

use of the term 'line' in order to formulate its *problem space* $\mathcal{P}$.

The phrase 'solution space' has been frequently used as though all solutions exist in one cohesive cluster, and in some contexts that may be a helpful conceptualization. But here, a definition is needed that can describe physical and functional characteristics of very different design ideas, so a more applicable notion of the solution space in this case is to consider each distinct concept as having its own *solution space* $\mathcal{S}$ where various embodiments exist on shared axes. Whereas an internal combustion engine and an electric motor may both be considered solutions for the same design problem, the solution space of the former might have axes such as number of pistons, bore diameter, and stroke length; while the latter would forego those axes and incorporate ones for number of windings, number of poles, and so forth.

The separation of solution spaces for each competing design concept imposes the challenge of comparing one to another in a useful way. However, designers already use criteria as a means for objective decision-making. As with constraints, these criteria can be used to form a third space called the *quality space* $\mathcal{Q}$ with axes that each idea under consideration will share and that helps to differentiate between them based on merits that address stake-holder needs. Driving range, for example, might be a point of comparison for a commuter vehicle that is agnostic to the type of propulsion it uses. In complement to that purpose, each criterion can be ascribed a weight so that relative importance can be considered. A utility score (or range of scores) is also put forth recognizing that the benefits of each criterion is not always linearly coupled to physical performance or characteristics.

Each of these spaces has an embedding space from which it inherits its axes and between which the maps operate. The embedding space for the problem is called the *constraint space* $\mathcal{C}$, the one for the solution is called the *form space* $\mathcal{F}$, and quality lies within the *objective space* $\mathcal{O}$. The maps are defined as operating on these larger embedding spaces to account for the fact that not every set of inputs or outputs for the maps will represent a solution to the design problem, so the embedding spaces represent the domains and ranges of the maps. The map from the form to the constraint space is called the *problem map* ${}^{p}\mathcal{M}$, and the one from the form to the objective space is the *quality map* ${}^{q}\mathcal{M}$. The two maps can be said to define the *design concept* they are associated with, $H = ({}^{p}\mathcal{M}, {}^{q}\mathcal{M})$. As will be shown subsequently, any changes to a premise or characteristic idea behind an approach to the problem will necessitate a change to one or both of its maps. Note that 'design concept' differs from the singular term 'design' in this document. A 'design' is a single

point in any of the three spaces, where each aspect of its parent concept has been instantiated with specific values. Furthermore, a design is not necessarily a solution to the given problem.

All of these constructs work in concert to fully describe a design problem and its solutions in a way that lends itself to objective, mathematical analysis and which can be incorporated into existing frameworks as well as other analytical methods. Figure 2.1 offers a graphical summary of the spaces proposed and the relations between them.

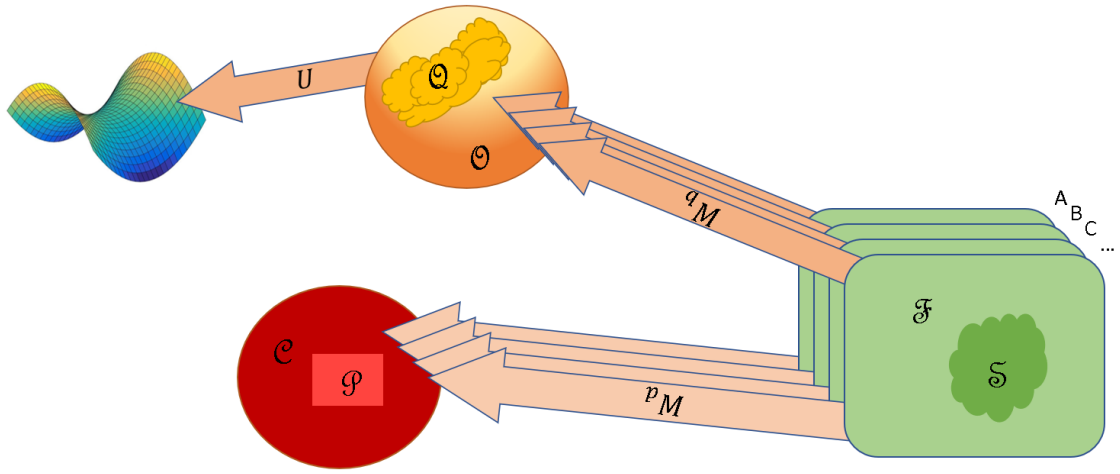

Figure 2.1: Basic relationship between design spaces and maps. Note that concepts may or may not share a solution space, but they will share a constraint and object space and they will not share maps.

2.1.1 Requirements

To begin, the requirements must be constructed in such a way as to retain their conceptual meaning while also providing the mathematical basis necessary for topological spaces. Let $\mathcal{A} = \{\alpha_1, \alpha_2, ..., \alpha_n\}$ be a set of parameters that must be constrained in order to meet the needs of the stakeholders and create a viable product. These are the constraint parameters for the problem and they are likely to change as the problem becomes better understood. As an example, if a project necessitated restrictions on volume V and mass m, then $\mathcal{A} = \{V, m\}$.

With these parameters identified, let $\mathcal{R} = \{\mathcal{R}_\alpha \; \forall \; \alpha \in \mathcal{A}\}$ be a set of numerical requirements, indexed by $\mathcal{A}$, so that there is a one-to-one correspondence between the set of constraint parameters and the set of requirements. Furthermore, let each requirement be defined as

$$\mathcal{R}_\alpha = \{y_\alpha \in \mathbb{R} \mid \text{stakeholder needs are met when } \alpha = y_\alpha, \alpha \in \mathcal{A}\}. \tag{2.1}$$

This definition characterizes each constraint $\mathcal{R}_\alpha$ as the set of all numerical values which the parameter α could take without violating any stakeholder needs. To continue the example above, $\mathcal{A} = \{V, m\}$ would imply that there is a requirement $\mathcal{R}_V$ for volume and another $\mathcal{R}_m$ for mass. If the problem necessitated an upper limit of 100 kg for mass, then $\mathcal{R}_m = \{x \mid 0 \text{ kg} < x \leq 100 \text{ kg}\}$. In this manner, each requirement may be composed of one or more continuous intervals or discrete values. When necessary, complex-valued constraints can be obtained by separating them into two requirements and representing the imaginary term as a real multiplier on i. Besides being numerical, a requirement must also be testable, meaning it should always be possible to determine whether a value is allowable or not. Excepting notation, these stipulations are not significantly different from much of the existing guidance for writing engineering requirements (INCOSE, 2015; Hirshorn et al., 2017). Additionally, certain topological constructs may make it possible to extend this definition to accept categorical, or non-numerical, requirements in the future.

2.1.2 Problem Space

The constraint parameters in $\mathcal{A}$ are also used to define the constraint space. To ensure it meets these three criteria, let the constraint space be defined as the pair $(\mathcal{C}, \tau_\mathcal{C})$ so that

$$\mathcal{C} = \prod_{\alpha \in \mathcal{A}} \mathbb{R} = \{y = (y_1, y_2, ..., y_n) \mid y_i \in \mathbb{R}, 1 \leq i \leq n\} \tag{2.2}$$

$$\tau_\mathcal{C} = \{\bigcup_{y \in \mathcal{C}} B(y, \epsilon_y) \mid y \in \mathcal{C},\ \epsilon_y \in \mathbb{R}^+\} \tag{2.3}$$

where $n = |\mathcal{A}|$ and

$$B(y, \epsilon_y) = \{y' \in \mathcal{C} \mid d(y, y') < \epsilon_y\} \tag{2.4}$$

where d is the distance function on the space chosen by the engineer. Any distance function meeting the criteria may be used, and if multiple topologies or metrics are desired, disparate spaces can be combined to form product spaces.

Since the axes of $\mathcal{C}$ are designated according to the constraint parameters in $\mathcal{A}$, each coordinate of a point $y \in \mathcal{C}$ corresponds to a value for the parameter associated with that axis. Certain points in $\mathcal{C}$ exist for which every coordinate value is a member of its respective requirement. In other words, they meet all of the requirements, and the problem space $\mathcal{P}$ is the collection of all of those points. This collection forms a subspace of $\mathcal{C}$ and can be formally defined as

$$\mathcal{P} = \prod_{\alpha \in \mathcal{A}} \mathcal{R}_\alpha = \{y = (y_1, y_2, ..., y_n) \mid y_i \in \mathcal{R}_{\alpha_i}, 1 \leq i \leq n\}. \tag{2.5}$$

2.1.3 Problem Map

To construct the solution space, a way of testing candidate designs against requirements is needed. Topologically, this means mapping between the spaces. Essentially, engineers already do this when determining whether a concept warrants further investigation or defining a plan for the verification and validation of their designs to ensure requirements are met. Here, these checks are gathered in their various forms into a set ${}^p\mathcal{M}$ defined as

$${}^p\mathcal{M} = \{f_\alpha : \mathcal{F} \to \mathcal{C}_\alpha \; \forall \; \alpha \in \mathcal{A}\} \tag{2.6}$$

where $\mathcal{C}_\alpha$ represents the 1-dimensional axis of $\mathcal{C}$ corresponding to the constraint parameter α and $\mathcal{F}$ is the form space mentioned at the beginning of the chapter and which will be formally defined in Section 2.1.4. Each element in ${}^p\mathcal{M}$ is a function corresponding to a specified constraint parameter α in $\mathcal{A}$, which outputs a value of α for each point in $\mathcal{F}$. Often, functions are thought of as mathematical equations, but in this context the term also includes other tools used by engineers to assess the adherence of a design to a constraint, such as optimization algorithms, machine learning models, or lookup tables. Once constructed, ${}^p\mathcal{M}$ may be treated as a single function ${}^p\mathcal{M} : \mathcal{F} \to \mathcal{C}$. In the context of a design space, this implies that every design $x \in \mathcal{F}$ maps to one and only one point $y \in \mathcal{C}$. However, in general, ${}^p\mathcal{M}$ is not assumed to be either surjective or injective; meaning some points in $\mathcal{C}$ may have two or more corresponding designs, while others may have none.

2.1.4 Solution Space

Just as the constraint parameters defined the axes of the constraint space, the input parameters for ${}^p\mathcal{M}$ likewise define the axes of the form space. Let the collection of the input parameters for

${}^{p}\mathcal{M}$ be referred to as the set $\mathcal{B} = \{\beta_1, \beta_2, ..., \beta_m\}$, where each $\beta \in \mathcal{B}$ represents a design parameter whose value may be independently chosen by the designer and m is the number of unique design parameters needed to calculate an output from each $f_\alpha \in {}^{p}\mathcal{M}$. Every member function of ${}^{p}\mathcal{M}$ will take some subset $\mathcal{B}' \subseteq \mathcal{B}$ as its inputs, and any individual β may serve as an input for multiple functions. The set $\mathcal{B}$ only contains those parameters which are essential to validate the current set of requirements and will likely not incorporate all of the parameters required for production until the end of the design process has been reached. The solution space can therefore be determined at any time during the design process provided ${}^{p}\mathcal{M} \neq \varnothing$, which also entails $\mathfrak{R} \neq \varnothing$. These parameters shall be elements of the design that can be independently and directly controlled, such as the physical dimensions of a component or the number of windings in a transformer. Parameters such as mass or volume would not, in general, be considered design parameters in this context, since their value is usually dependent on other independent parameters.

Let the embedding space $\mathcal{F}$ be defined as $(\mathcal{F}, \tau_\mathcal{F})$ such that

$$\mathcal{F} = \prod_{\beta \in \mathcal{B}} \mathbb{R} = \{x = (x_1, x_2, ..., x_m) \mid x_i \in \mathbb{R}, 1 \leq i \leq m\} \tag{2.7}$$

$$\tau_\mathcal{F} = \{\bigcup_{x \in \mathcal{F}} B(x, \epsilon_x) \mid x \in \mathcal{F},\ \epsilon_x \in \mathbb{R}^+\} \tag{2.8}$$

where $m = |\mathcal{B}|$. As with $\mathcal{C}$ before, $\mathcal{F}$ is also equivalent to the real space but with a different number of dimensions from $\mathcal{C}$. Every point $x \in \mathcal{F}$ maps to a corresponding point in $\mathcal{C}$ according to $y = {}^{p}\mathcal{M}(x)$. Then let the solution space be

$$\mathcal{S} = {}^{p}\mathcal{M}^{-1}(\mathcal{P}) = \prod_{\alpha \in \mathcal{A}} f_\alpha^{-1}(\mathcal{P}) = \{x \in \mathcal{F} \mid {}^{p}\mathcal{M}(x) \in \mathcal{P}\}. \tag{2.9}$$

That is to say, $\mathcal{S}$ is the inverse image of $\mathcal{P}$ under ${}^{p}\mathcal{M}$. Equivalently, $\mathcal{S}$ can be described as all points in $\mathcal{F}$ that map to $\mathcal{P}$ via ${}^{p}\mathcal{M}$. The space $\mathcal{S}$ must be defined according to this inverse relationship owing to the assumption that ${}^{p}\mathcal{M}$ may not be surjective. Conceptually, $\mathcal{S}$ is a collection of all of the possible designs that are considered viable solutions to the given engineering problem, according to the design decisions that have been made.

2.1.5 Quality Space

During any engineering design project, it will typically be necessary to compare potential designs that are quite different from one another. The form space, as it has been defined here, is able to capture any physical description, but it does not offer a way to directly compare designs with differing fundamental structures since they would exist along disparate axes. For such a comparison to take place, a space is needed in which all relevant designs inhabit a shared set of axes.

To that end, an *objective space* $\mathcal{O}$ is defined that allows for a direct comparison between dissimilar solution spaces. The axes of this new space correspond to a set of generic characteristics that are agnostic to the form of the design and can grade solutions according to some specified benefit. This set of objective parameters can be represented as

$$\Gamma = \{\gamma_1, \gamma_2, ..., \gamma_p\} \tag{2.10}$$

where each $\gamma \in \Gamma$ represents some aspect of the overall design that is ubiquitous among different designs, perhaps some performance metric or desirable characteristic. Each γ should provide some basis for distinguishing between designs that can be used to aid in objective decision-making. Associated with each parameter is a metric space $\mathcal{O}_\gamma = (X_\gamma, d_\gamma)$ where X_γ is comprised of all possible values that its associated γ may assume and d_γ is the distance function for the space. From these, the space $\mathcal{O}$ can be defined as the product space

$$\mathcal{O} = \prod_{\gamma \in \Gamma} \mathcal{O}_\gamma \tag{2.11}$$

having the metric

$$d_O(x, y) = \Big(\sum_{\gamma \in \Gamma} d_\gamma\big(x_\gamma, y_\gamma\big)\Big)^{\frac{1}{2}} \tag{2.12}$$

for $x, y \in \mathcal{O}$.

Then a quality map ${}^q\mathcal{M} : \mathcal{F} \to \mathcal{O}$ is defined, similar to ${}^p\mathcal{M}$ previously, that maps points in the form space into the objective space, with exactly one mapping function for each criterion.

$${}^q\mathcal{M} = \{f_\gamma : \mathcal{F} \to \mathcal{O}_\alpha \ \forall \ \gamma \in \Gamma\} \tag{2.13}$$

As with ${}^{p}\mathcal{M}$, this map is associated with a specific design concept. So there may be numerous maps for a given design project when multiple concepts are being considered. Given this map, the quality space $\mathcal{Q}$ for a specific concept is defined as the image of its solution space in $\mathcal{O}$ under its quality map. More compactly,

$$\mathcal{Q} = {}^{q}\mathcal{M}(\mathcal{S}). \tag{2.14}$$

Additionally, these criteria will each have a utility curve $u_\gamma : \mathcal{O}_\gamma \to \mathbb{R}$ that specifies the merit of each point along the axis. Parameters will also have a weight w_γ that denotes its importance relative to other parameters in the objective space. A response surface U can then be created by the equation

$$U(x) = \sum_{\gamma \in \Gamma} w_\gamma u_\gamma(x_\gamma) \tag{2.15}$$

for $x \in \mathcal{O}$. With this response surface, one can now find the weighted utility score of any point in the objective space according to the needs of the designer. These definitions for $\mathcal{O}$ and U provide a means for designers to directly compare arbitrarily disparate design concepts from a topological standpoint. They can also determine not only which concepts offer the single highest quality solution but also the distribution of quality over the space.

2.1.6 Design Concepts

It will be necessary in most instances to distinguish between various design concepts as well as between iterations of the constructs and designs themselves. In this paper, the symbol H will be used to refer to a design concept as a whole. This amounts to the collection of all design decisions that are being considered in concert. Since these decisions are encapsulated in the two map sets, one may say that a concept is *defined* by its pair of maps that is $H_a = ({}^{p}\mathcal{M}, {}^{q}\mathcal{M})$. Therefore, in order for two concepts to be considered distinct, at least one element of one of their map sets must differ.

2.2 Design Space Construction

A simple example is used to illustrate the process of obtaining these spaces, from requirements definition through solution space visualization. An early stage in the design process may be assumed where only two constraints have been placed on the design problem:

1. The volume of the product must be between 25 and 50 m^3, inclusive.
2. The mass of the product must be between 20,000 and 60,000 kg, inclusive.

First, determine the parameters being constrained to establish the set $\mathcal{A}$ and define the appropriate requirement sets. In this case, volume V and mass m are the constraint parameters. So

$$\mathcal{A} = \{V, m\} \tag{2.16}$$

$$\mathcal{R}_V = \{x \mid 25\mathrm{m}^3 \leq x \leq 50\mathrm{m}^3\} \tag{2.17}$$

$$\mathcal{R}_m = \{x \mid 20,000\mathrm{kg} \leq x \leq 60,000\mathrm{kg}\} \tag{2.18}$$

$$\mathcal{R} = \{\mathcal{R}_V, \mathcal{R}_m\}. \tag{2.19}$$

The resulting constraint space $\mathcal{C}$ is the 2-dimensional real space with m along one axis and V on the other. Within $\mathcal{C}$, the Cartesian product $\mathcal{R}_m \times \mathcal{R}_V$ provides the problem space $\mathcal{P}$. Figure 2.2 depicts $\mathcal{P}$ as the shaded region residing within the plane of $\mathcal{C}$.

To develop the map and solution space, design decisions need to be made. These can be changed later without sacrificing analyzability, if desired. Selecting a solid sphere as the initial design concept gives a map set ${}^{p}\mathcal{M}$ as follows:

$$f_V = \frac{4}{3}\pi r^3 \tag{2.20}$$

$$f_m = \frac{4}{3}\rho\pi r^3 = \rho f_V(r) \tag{2.21}$$

$${}^{p}\mathcal{M} = \left\{ \begin{array}{c} f_m(r) \\ f_V(\rho, r) \end{array} \right\}. \tag{2.22}$$

From ${}^{p}\mathcal{M}$, a set of design parameters is obtained that can be directly manipulated when

designing the product, the density ρ and radius r of the sphere. This provides the design parameter set $\mathcal{B} = \{\rho, r\}$ and gives $\mathbb{R}^2$ as the form space, with ρ and r as the axes. In a simple case, such as this, it is relatively simple to analytically determine the solution space, leading to the plot in Figure 2.3.

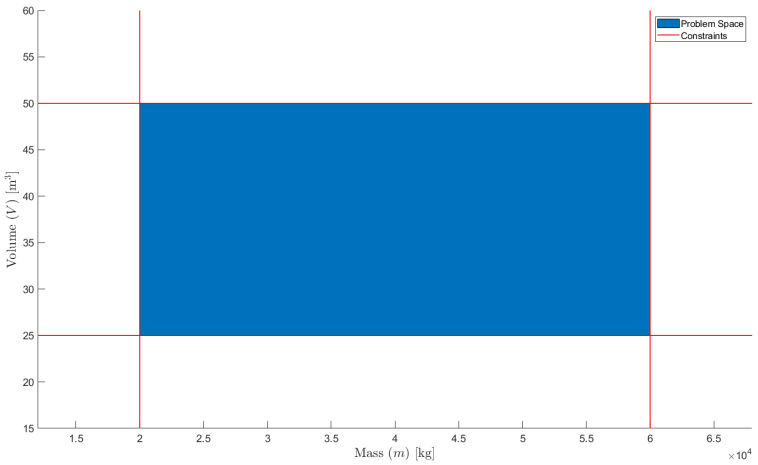

Figure 2.2: Lines show the bounds on constraint parameters, shaded region depicts problem space.

Figures 2.4 and 2.5 demonstrate how the solution space gets its shape by plotting each constraint parameter against its associated design variables. The projection of the surface in Figure 2.4 and that of the curve in Figure 2.5 each bound the shaded region in Figure 2.3 to form the solution space for this problem under the given design decisions.

In many cases, it will not be feasible or even possible to analytically determine the solution space, as was done in this example. In those instances, the solution space can be approximated via sampling. To do this, the engineer must use their judgment to determine an appropriate sampling method, search space, and sample size. The approach would change depending on the space and requirements, but here a reasonable search space can be obtained by rearranging the equation for volume for r and solving for the radius at the upper and lower bounds of $\mathcal{R}_V$, giving the search space for r. Then finding the upper search bound for ρ by dividing the highest allowable mass by the lowest volume, and the lower one using the lowest mass divided by the highest volume. Figure 2.6 shows another important consideration for the sample-based approach, which is the difference in the resolution of the solution space obtained from various sample sizes.

To construct the quality space for this example, criteria need to be specified against which

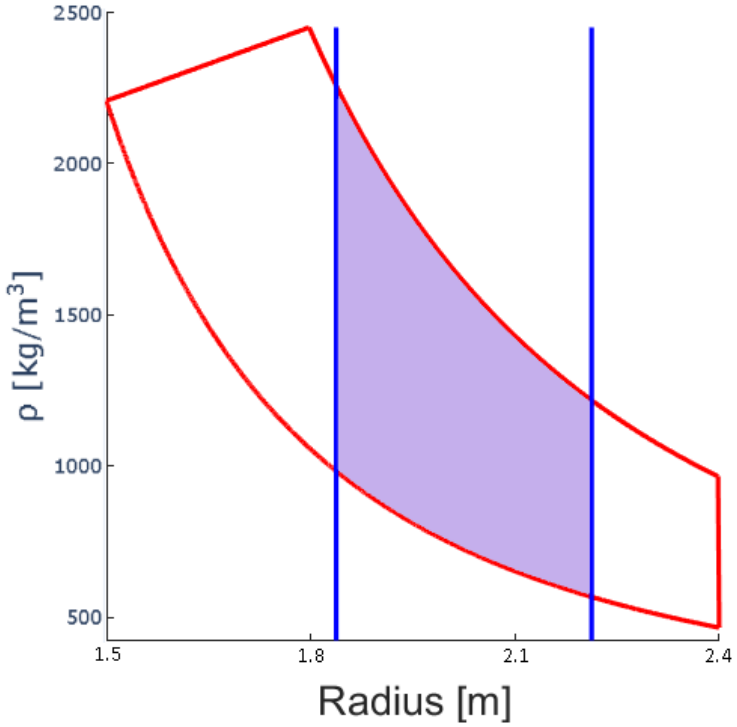

Figure 2.3: Shaded region is the solution space.

various designs can be compared. Here, buoyancy and Reynolds number will be used so that the interactions of complex curves may be demonstrated in low dimensions. Suppose that neutral buoyancy is desired along with a high Reynolds number. A line similar to a normal distribution curve might be used to incentivize near-neutral buoyancies, as a fraction of the overall weight of the object. And a curve similar to the inverse Reynolds-Drag Coefficient curve can be used to incentivize higher Reynolds numbers. Note that these are merely examples and the designer is at liberty to assign utility curves according to the needs of their project. As with mapping functions, curves need only associate output values to input values and need not arise from mathematical formulae if another method suits the project better. Figure 2.7 shows the selected curves.

These curves can then be scaled, weighted, and combined to form a utility response surface. Since the utility score is based on shared attributes, it shifts the focus for comparison toward how well the design meets the needs of the problem rather than the form it takes. Figure 2.8 depicts the surface obtained from the utility curves above, with a weight of 2 applied to the buoyancy and 3 applied to the Reynolds number.

With the criteria and utility response defined, the quality space may be constructed. For

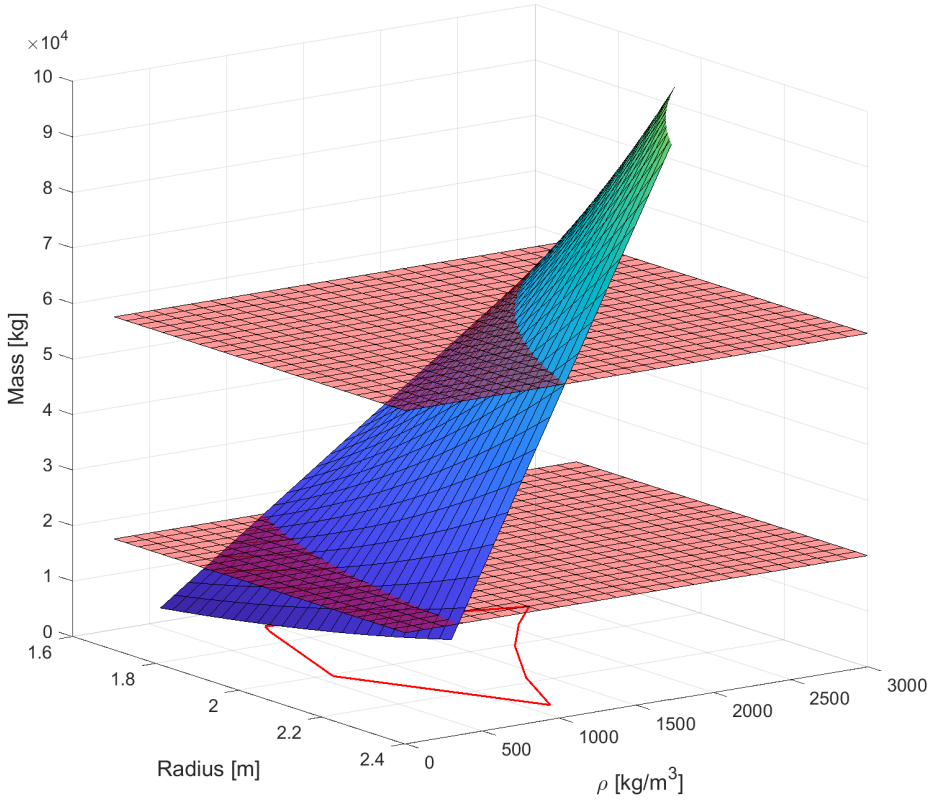

Figure 2.4: Outline at the base of the plot is the projection of the portion of the surface which lies between the min and max allowable mass.

the physical properties of water and the Reynolds equation, it is assumed that the object will be submerged in freshwater, at 20°C moving with an average relative velocity of 5 m/s. The quality map for this example then becomes

$$f_{buoyancy} = \frac{\rho_{water} - \rho}{\rho} \tag{2.23}$$

$$f_{Reynolds} = \frac{\rho_{water} * u * 2 * r}{\mu_{water}} \tag{2.24}$$

$${}^{q}\mathcal{M} = \left\{ \begin{array}{c} f_{buoyancy}(\rho) \\ f_{Reynolds}(r) \end{array} \right\} \tag{2.25}$$

where u is the anticipated relative free-stream velocity of water and μ_{water} is the dynamic velocity of freshwater.

Using ${}^{q}\mathcal{M}$, points in the solution space are mapped into the objective space to form the

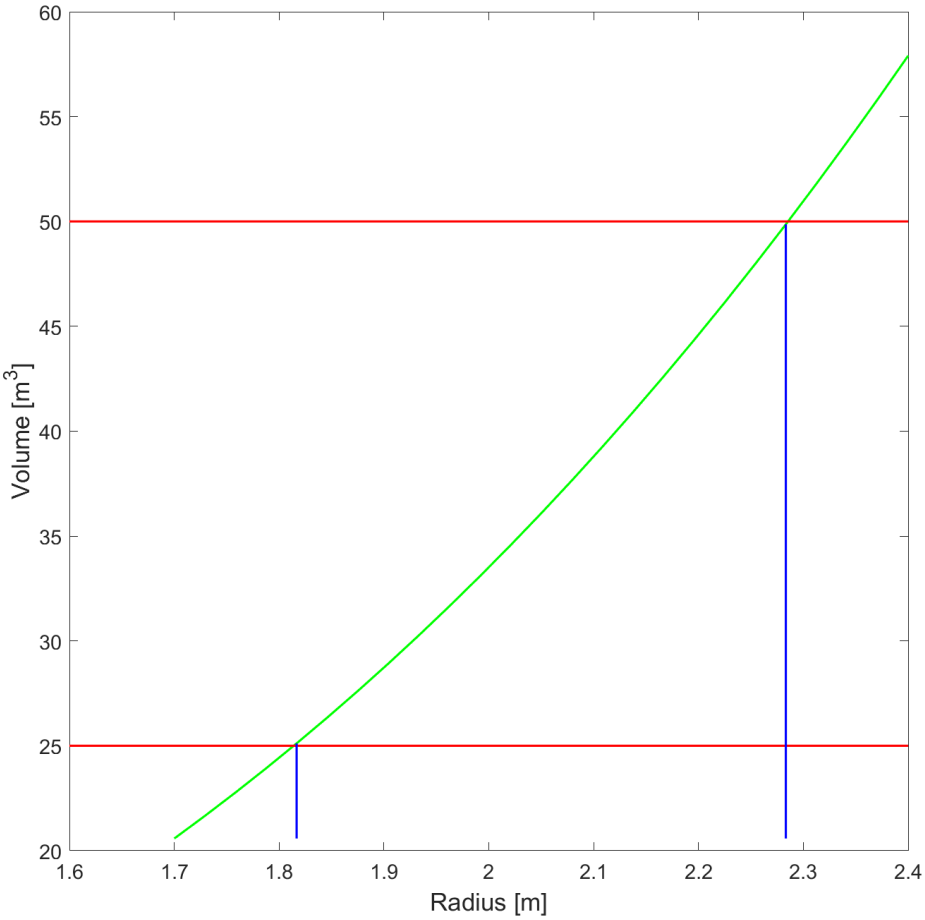

Figure 2.5: The vertical lines show the projection of the curve between the min and max volume onto the horizontal.

quality space for the design. The utility response is applied as a color gradient in Figure 2.9 so the the relative performance of each solution can be visualized.

Similarly, by tracking which points in the quality space correlate to those in the solution space along with their individual utility values, the gradient can be retroactively applied to the solution space as well. This is shown in Figure 2.10.

2.3 Design Space Analysis

Having laid the topological foundations for the design spaces, various applications can be explored. This section will focus on three potential applications of these techniques: similarity, sensitivity, and conformity.

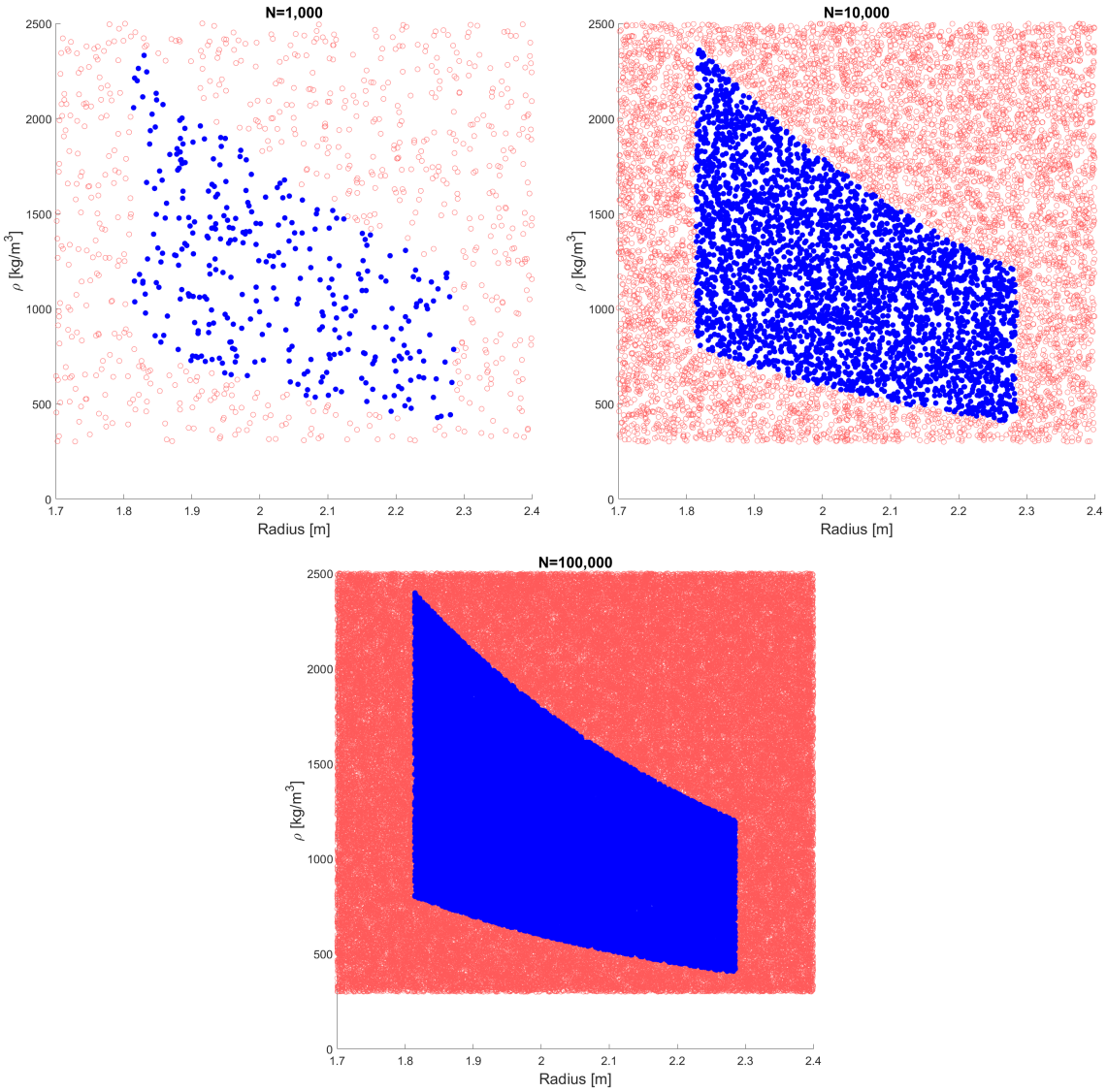

Figure 2.6: The same solution space as previously shown, but constructed through sampling techniques. By N = 100,000, a fairly high-resolution image of the solution space boundaries can be seen.

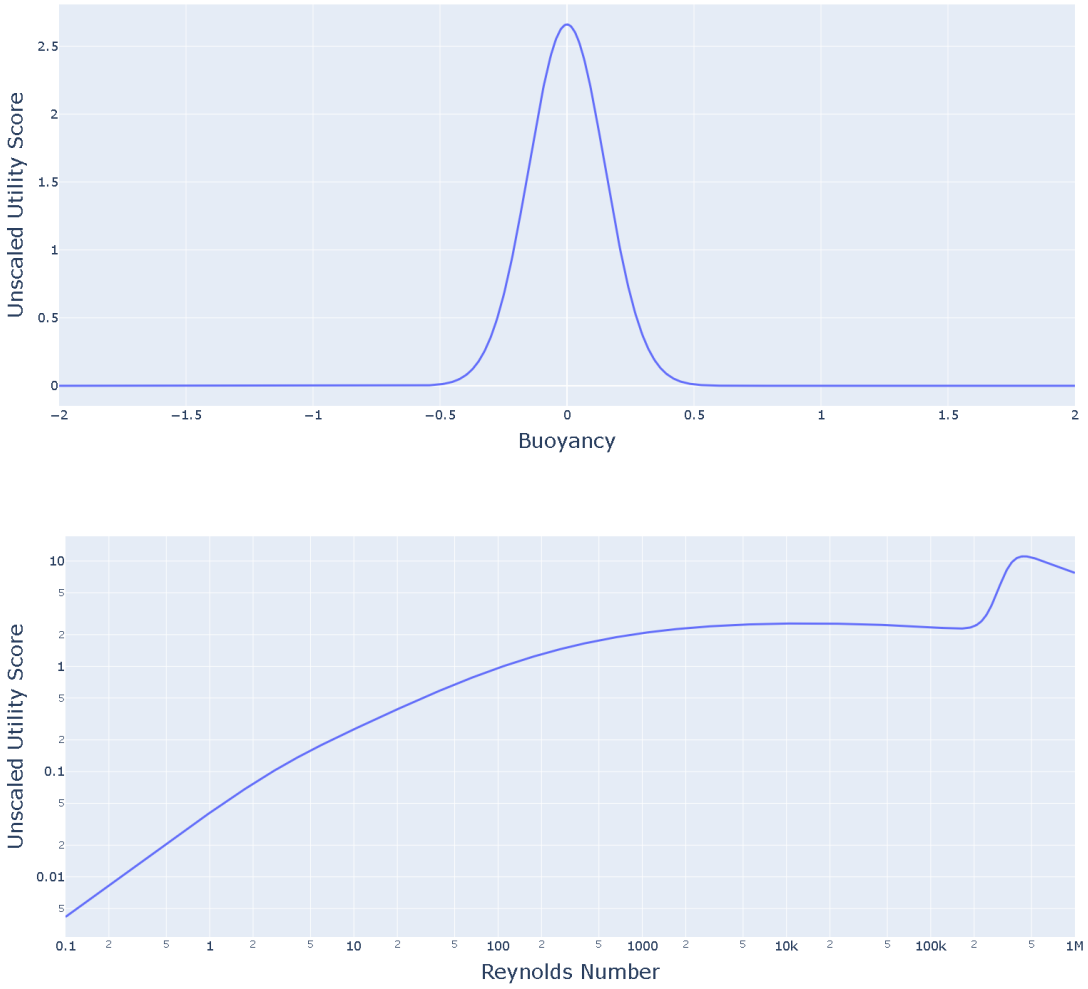

Figure 2.7: Utility curves for the chosen criteria

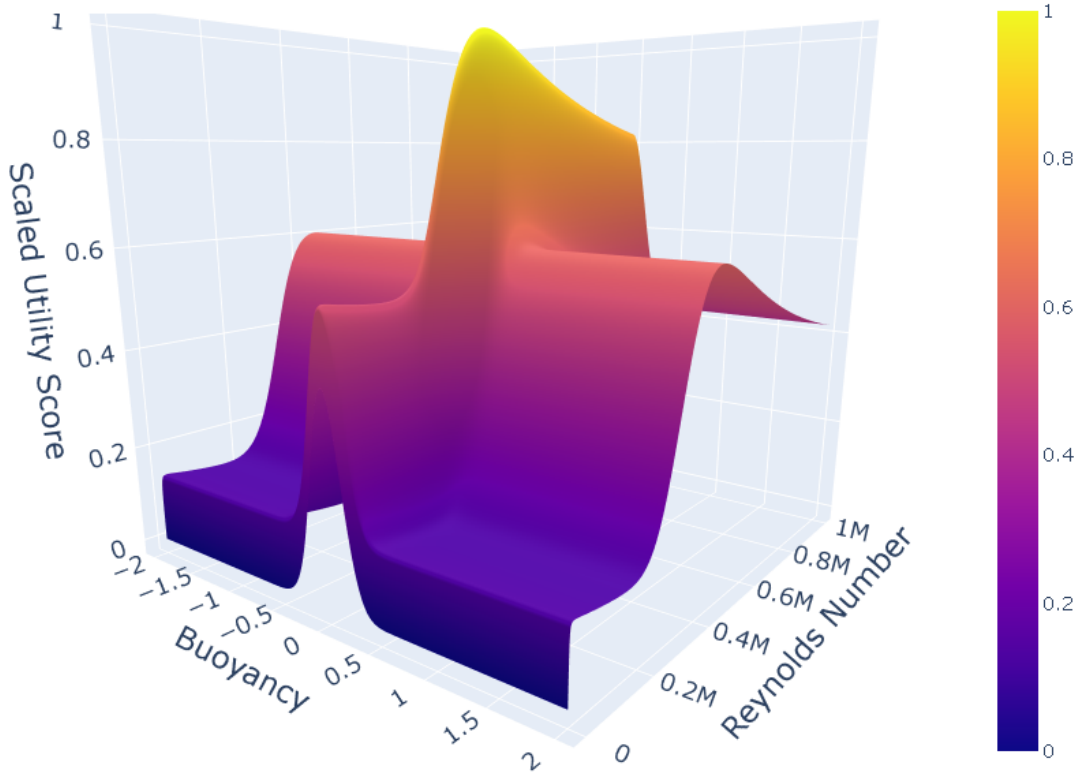

Figure 2.8: Response surface arising from the linear combination of utility curves. A weight of 2 has been applied to the buoyancy curve and 3 to the Reynolds number curve.

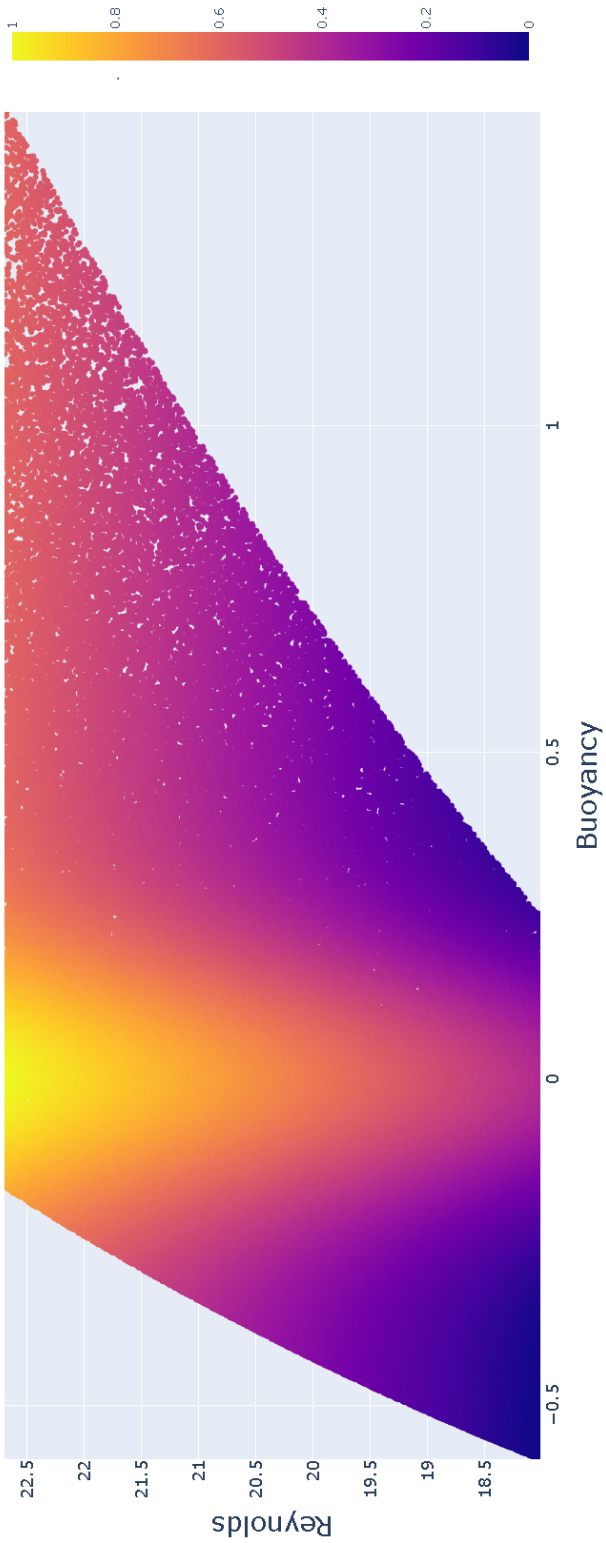

Figure 2.9: Design concept quality space with utility gradient applied.

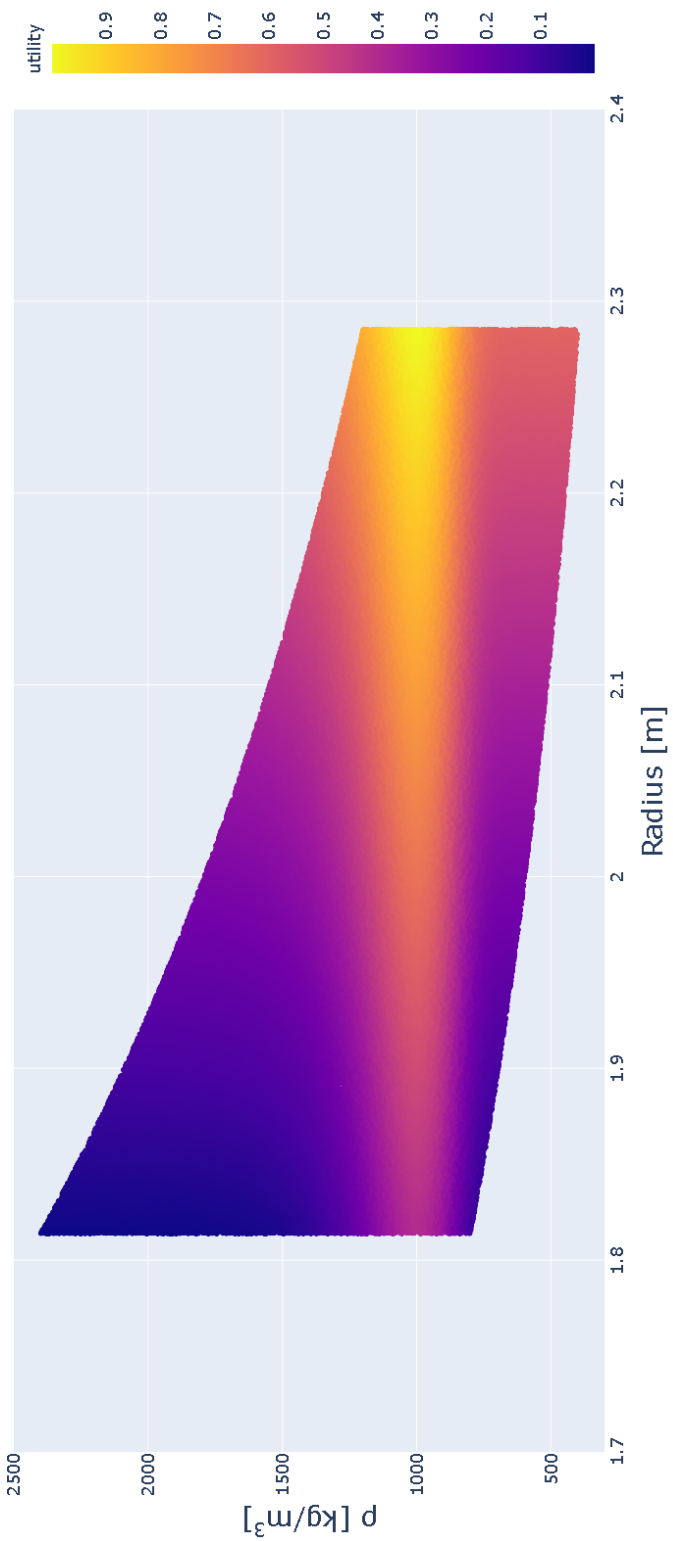

Figure 2.10: Design concept solution space with utility gradient applied.

2.3.1 Similarity Between Like Spaces

There are a variety of similarity measures available for comparing discrete sets. Two of the most common, the Jaccard index (Jaccard, 1912) and the Overlap Coefficient (Simpson, 1943), are respectively defined as

$$\mathrm{oc}(X,Y) = \frac{|\ X \cap Y\ |}{\min(|\ X\ |, |\ Y\ |)} \tag{2.26}$$

$$J(X,Y) = \frac{|\ X \cap Y\ |}{|\ X \cup Y\ |}. \tag{2.27}$$

Since design spaces may contain an infinite number of points, these indices can be extended as needed to accommodate any topological measures μ for size beyond simple cardinality. In cases where analytical measurement is not possible or practical, the measure can be approximated by sampling increasingly large subsets until convergence. Letting the representative samples of a set be denoted by

$$\breve{X} = \{x_i \mid x_i \in \Omega_X, i = 1, ..., N\} \tag{2.28}$$

where Ω_X is the sample space of X and N is the number of points in the sample. Then the indices can be modified as

$$\tilde{J}(X,Y) = \frac{\mu(X \cap Y)}{\mu(X \cup Y)} \approx \frac{|\ \breve{X} \cap \breve{Y}\ |}{|\ \breve{X} \cup \breve{Y}\ |} \tag{2.29}$$

$$\tilde{\mathrm{oc}}(X,Y) = \frac{\mu(X \cap Y)}{\min(\mu(X), \mu(Y))} \approx \frac{|\ \breve{X} \cap \breve{Y}\ |}{\min(|\ \breve{X}\ |, |\ \breve{Y}\ |)} \tag{2.30}$$

both of which have a range of [0, 1]. The respective values given by these formulas provide different information about the sets or spaces in question. While a value of zero indicates disjoint sets for both measures – assuming finite cardinality, $J(X,Y) = 1$ indicates $A = B$ whereas $\mathrm{oc}(X,Y) = 1$ signifies either $A \subseteq B$ or $B \subseteq A$.

For the purposes of design, both of these measures convey information that can help the designer understand the commonality of two spaces. However, they also have weaknesses. The Jaccard index fails to differentiate between situations where the sizes of the individual sets vary

but the sizes of the union and intersection remain the same. Overlap, on the other hand, cannot distinguish any changes in relative set size when one set is a subset of the other. Figure 2.11 demonstrates these circumstances graphically.

Due to their limitations, a combination of these equations is proposed that will more appropriately quantify similarity for use in design spaces. Let

$$\text{ss}(X,Y) = \frac{\sqrt{\tilde{J}(X,Y)^2 + \tilde{oc}(X,Y)^2}}{\sqrt{2}}. \tag{2.31}$$

This new equation, referred to as spatial similarity ss, still offers a range of [0,1] with ss = 0 indicating no similarity and ss = 1 if and only if $A = B$, as with the Jaccard index. However, it also captures differences in relative size when neither of the other two are able to. Table 2.1 illustrates how the three formulas handle ambiguous cases.

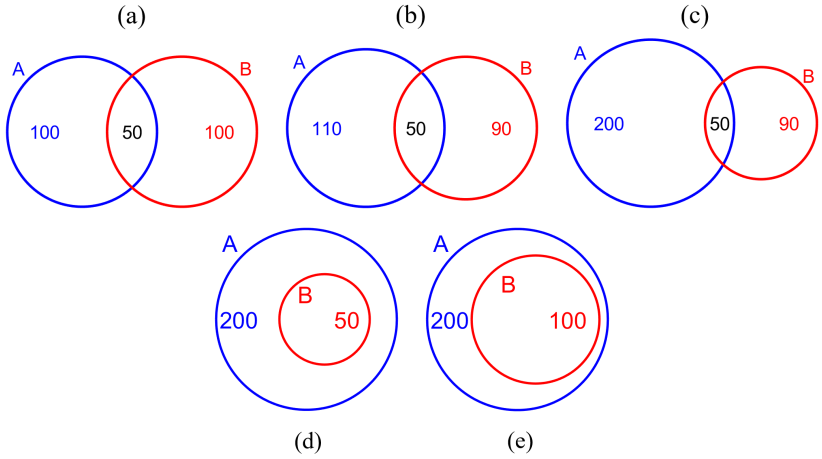

Figure 2.11: Examples of set relations that Jaccard and Overlap have difficulty differentiating. Numbers indicate the size of the space given by $\mu(X)$. The size of the intersections is included in both set sizes.

Table 2.1: Similarity values for sets in Figure 4. Bolded values denote similarity scores shared with another case for the same index.

	$\tilde{J}$	$\tilde{oc}$	ss
a	**0.333**	0.500	0.425
b	**0.333**	**0.556**	0.458
c	0.208	**0.556**	0.420
d	0.250	**1.000**	0.729
e	0.500	**1.000**	0.790

It should be noted that certain choices of size measure μ may introduce scenarios wherein all of the similarity equations discussed result in a value of 0 when the intersection is in fact non-empty. For example, using volume gives a similarity of 0 when the intersection is a lower dimension than the spaces themselves, as in circles intersecting at a point or cubes intersecting on a face. However, it is generally expected that these occasions are rare and can be readily checked, as it would be indicated by all points in the intersection sharing the same value for at least one coordinate. Figure 2.12 illustrates the similarity between two versions of the example space from above, the original and one resulting from a change to the allowable range for each requirement.

Similarity can be quantified for problem spaces as well. In the case of simple, rectilinear spaces such as the one in Figure 2.2 the similarity may be directly calculated using the overlapping area of the two spaces. However, when such calculations become tedious or impractical, values may be randomly chosen from each space's respective set $\mathcal{R}$ and compiled into ordered sets of coordinates within each problem space. The two resulting collections of points can be considered as discretized representations of their associated problem spaces and then compared as described above for the solution space. As in the previous case, the accuracy of the resulting similarity value will improve as the number of selected points increases. One complication that may arise is the case where one or more of the problem space bounds extends to infinity. One method for handling this situation will be explained in Chapter 3

2.3.2 Sensitivity to Change

Since the similarity indices presented in Section 2.3.1 all have ranges on the interval [0,1], it is possible to quantify change in a space X from state 1 to state 2 as

$$\Delta X_{1\to 2} = 1 - \text{Similarity}(X_1, X_2). \tag{2.32}$$

It should be noted that any similarity index that maps to that same interval may be substituted in this equation. This concept of change can then be used to define a new measure of sensitivity between related spaces, such as the sensitivity of the solution space to changes in the problem space. Since $\mathcal{S}$ is dependent on $\mathcal{P}$, their states are intrinsically linked. And the sensitivity of that link can be quantified by

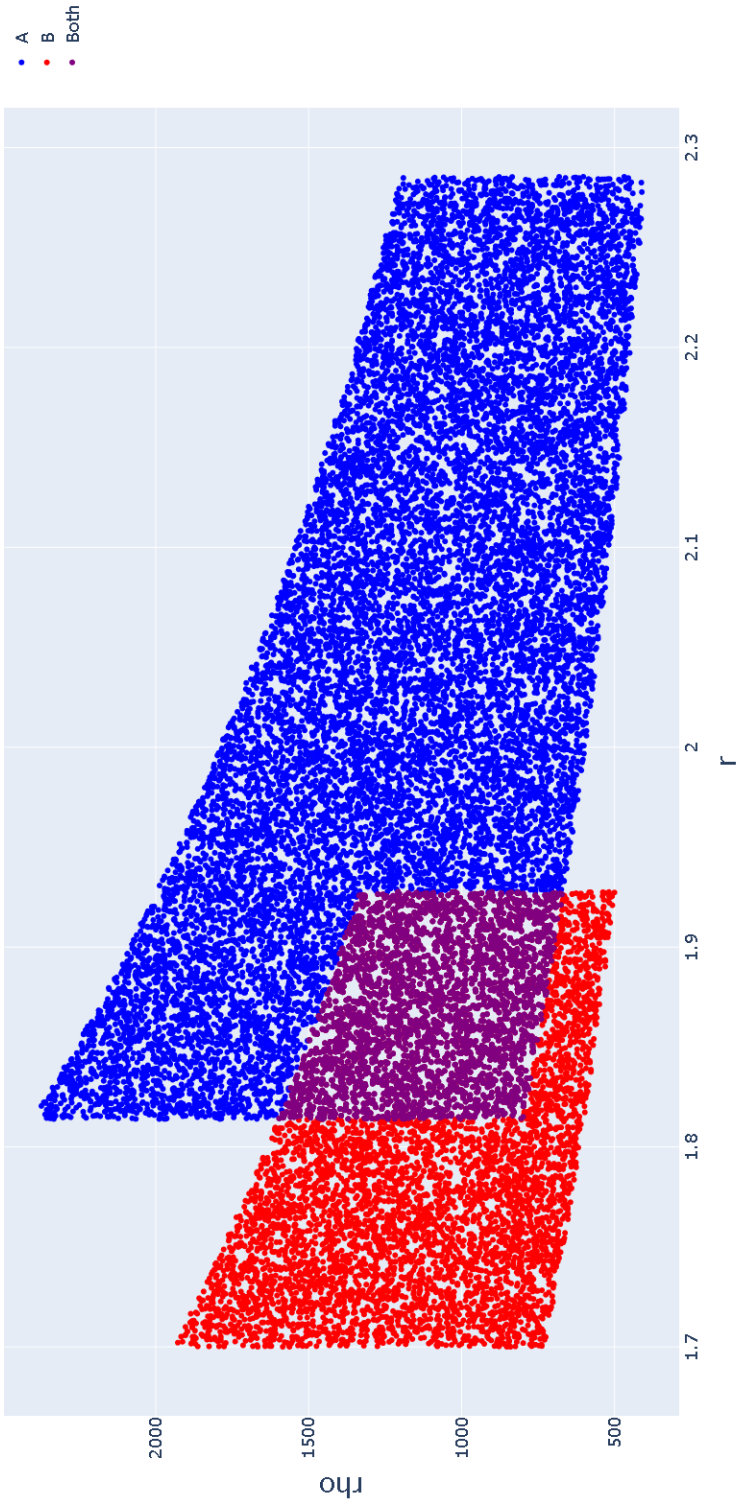

Figure 2.12: Shows the similarity of the solution space from Figure 2.3 to one that would result from adjusting $\mathcal{R}_V$ to [15m^3, 30m^3] and $\mathcal{R}_m$ to [15,000kg/m^3, 40,000kg/m^3]. The calculated similarity of the spaces shown is 0.269.

$$\frac{\Delta \mathcal{S}}{\Delta \mathcal{P}} = \frac{1 - \text{Similarity}(\mathcal{S}_1, \mathcal{S}_2)}{1 - \text{Similarity}(\mathcal{P}_1, \mathcal{P}_2)}. \tag{2.33}$$

This idea can be extended to determine the impact of each individual requirement on the solution space as well. Since each $\mathcal{R}_\alpha$ is a set and the similarity indices are set-based equations, the sensitivity measure can be used to compare $\mathcal{S}$ to $\mathcal{R}_\alpha$ for any requirement in $\mathcal{R}$. By extension, a notion of change gradient can also be determined such that

$$\bigtriangledown \mathcal{S} = \begin{bmatrix} \frac{\Delta \mathcal{S}}{\Delta \mathcal{R}_{\alpha_1}} \\ \vdots \\ \frac{\Delta \mathcal{S}}{\Delta \mathcal{R}_{\alpha_n}} \end{bmatrix}, \; n = \mid \mathcal{A} \mid \tag{2.34}$$

which can provide an indication of those requirements, or combinations of requirements, which would result in the greatest change in the solution space.

2.3.3 Conformity of Design to Constraints

The third technique is a measure of the conformity of a design, which indicates the smallest adjustment necessary to bring an infeasible design into the solution space. That is, the distance from a point in $\mathcal{F}$ which is not a member of $\mathcal{S}$ to the nearest boundary of $\mathcal{S}$.

The specific distance metric used can be tailored to the topology of the form space as well as to the needs of the engineer. In the example presented in Section 2.2, we have $\mathbb{R}^2$ as our form space and the Euclidean distance as our metric. Determining the nearest point in the solution space analytically may not be a trivial task in most cases. However, when sampling, there are a number of algorithms for finding nearest neighbors which may be used. Figure 2.13 demonstrates how this could be done for an infeasible design in the previously defined form space.

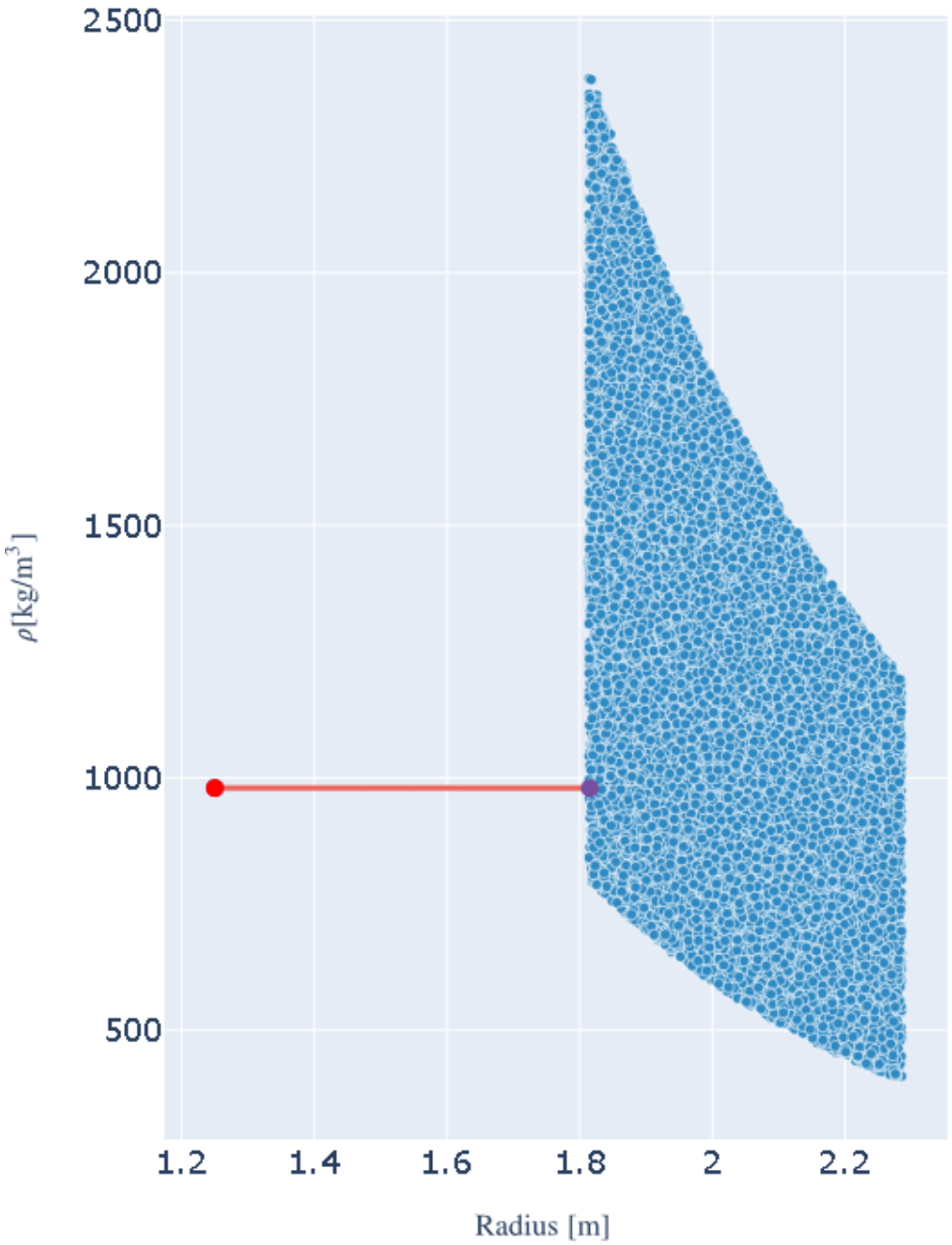

Figure 2.13: Minimum adjustment to bring a design having $r = 1.25$m and $\rho = 980$kg/m^3 into conformity with the requirements.

Chapter 3

Example Case

In this chapter, a more complex example will be used that will help to illustrate some of the ways this formalism can be employed in a more realistic design scenario. The example will cover the construction and visualization of higher-dimensional spaces as well as the use of these spaces to make adjustments and then evaluate the effects of those decisions. Furthermore, two different design concepts will be evaluated against a single problem and compared to one another using the methods proposed in the preceding chapter.

The example case is an excerpted and modified version of an actual design problem from another research effort (Roberts, 2021). A relatively small subset of the requirements from that project will be adapted for use in this demonstration. These requirements were selected from the project based on their ability to illustrate a variety of the characteristics of this formalism while keeping the dimensionality reasonable for visualization and calculation purposes. They are not intended to portray a complete set of the requirements necessary for comprehensive product design. Additionally, the map functions below are included to allow the reader to recreate the results and visualizations shown in this paper, they too have been simplified for demonstration purposes. It is hoped that the reader may see from this depiction the potential to scale up the proposed methods for use in larger projects having a full complement of requirements and criteria, where visualization of the complete space is not generally feasible.

3.1 Design Task

The design problem addressed in Roberts (2021) is to develop a 3D printer for use in a laboratory setting that can control as much as possible the parameters affecting part quality in fused-deposition modeling for additive manufacturing. Such a task requires, among other things, the ability of the device to control the resolution, excess motion, travel, and speed of the print head during the printing process. The requirements chosen for this paper from among the full project set address these aspects of the problem. They constrain the deformation in the printer components to guard against unintended movement, the speed and resolution that the print head is capable of achieving, and the horizontal cross-section of the build volume. Also, criteria are used to bias the design process toward large build areas, high print resolution, and a desired range of fast-travel (non-printing) speeds.

The constraints to be adapted for this example are as follows:

1. The frame must not deflect further than 0.005 inches between its upper and lower surfaces in the x-direction as a result of print head acceleration forces.

2. The frame must not deflect further than 0.005 inches between its upper and lower surfaces in the y-direction as a result of print head acceleration forces.

3. The gantry bar must withstand lateral deflection greater than 0.001 inches from its center-line at the mid-point as a result of print head acceleration forces.

4. The structure must allow for the nozzle to travel 8 inches or more in the x-direction.

5. The structure must allow for the nozzle to travel 8 inches or more in the y-direction.

6. The lateral motion system for the print head must support a fast-travel (non-printing) velocity of 16 in/sec or greater.

7. The lateral motion system for the print head must support a nozzle movement resolution of 0.001 in or less.

3.2 Application

3.2.1 Problem

To apply the formalism described in Chapter 2, the requirements must first be restated in point-set form so that values from the map function can be tested for membership. The index set $\mathcal{A}$ over which the requirement sets are indexed is created by assigning a symbol to each constrained parameter, as shown here:

- δ_x: x-direction deformation of upper frame
- δ_y: y-direction deformation of upper frame
- δ_G: Lateral deformation of gantry bar
- D_x: Nozzle x-travel
- D_y: Nozzle y-travel
- v: Fast-travel velocity of the print head
- $\Delta\ell$: Lateral resolution of the nozzle

$$\mathcal{A} = \{\alpha_1, ..., \alpha_n\} = \{\delta x, \delta y, \delta G, Dx, Dy, v, \Delta\ell\},\ n = 7 \tag{3.1}$$

Then, a requirement $\mathcal{R}_\alpha$ is created for each $\alpha \in \mathcal{A}$. These contain the allowable values for each requirement according to the text statements above.

- $\mathcal{R}_{\delta_x} = \{x \mid |x| \leq 0.001in\}$
- $\mathcal{R}_{\delta_y} = \{x \mid |x| \leq 0.005in\}$
- $\mathcal{R}_{\delta_G} = \{x \mid |x| \leq 0.001in\}$
- $\mathcal{R}_{D_x} = \{x \mid x \geq 8.0in\}$
- $\mathcal{R}_{D_y} = \{x \mid x \geq 8.0in\}$
- $\mathcal{R}_v = \{x \mid x \geq 16\ in/s\}$

- $\mathcal{R}_{\Delta\ell} = \{x \mid x \leq 0.001in\}$

$$\mathcal{R} = \{\mathcal{R}_{\delta_x}, \mathcal{R}_{\delta_y}, \mathcal{R}_{\delta_G}, \mathcal{R}_{D_x}, \mathcal{R}_{D_y}, \mathcal{R}_v, \mathcal{R}_{\Delta\ell}\} \tag{3.2}$$

The problem space is the Cartesian product of each requirement set. Since the seven requirements presented all have continuous intervals for their allowable values, the resulting space will be a filled volume in n-dimensional space, with $n = 7$. And since not all axes of the resulting space can be displayed in a single plot, Figure 3.1 shows the space with axes plotted pairwise to allow for visualization. It should be noted that these subplots are not slices but rather each is a projection of the entire space onto the plane of the subplot. So, while every point the space is present in every subplot, the perception of depth is lost and points appearing near to each other in the figure may in fact be separated along another axis.

As was done with the requirements, the criteria are also rewritten in point-set format. To distinguish more favorable designs according to the characteristics mentioned in Section 3.1, the resolution, print area, and fast-travel speed will be used as criteria, giving the axes of the objective space and the following criteria set:

$$\Gamma = \{\Delta\ell, P, s\} \tag{3.3}$$

For each of these, a curve is assigned that describes how utility value is distributed over the range of possible values for that criterion. They take the value of their associated criterion as input and return the design utility gained from that value.

$$u_{\Delta\ell} = \exp(-445\Delta\ell) \tag{3.4}$$

$$u_a = \tanh(a^{1.75} \times 10^5) \tag{3.5}$$

$$u_v = \max\left(0,\ \frac{100}{32.1\sqrt{2\pi}}\, e^{\frac{-(x-20)^2}{2(32.1^2)}} - \frac{7.3750}{x} + 0.056\right) \tag{3.6}$$

The utility curves for these criteria are shown in Figure 3.2. Note that the curves are intentionally designed to output values on the interval [0, 1] for the expected input interval. While not strictly necessary, doing this ensures equal representation of all criteria, bypassing the need

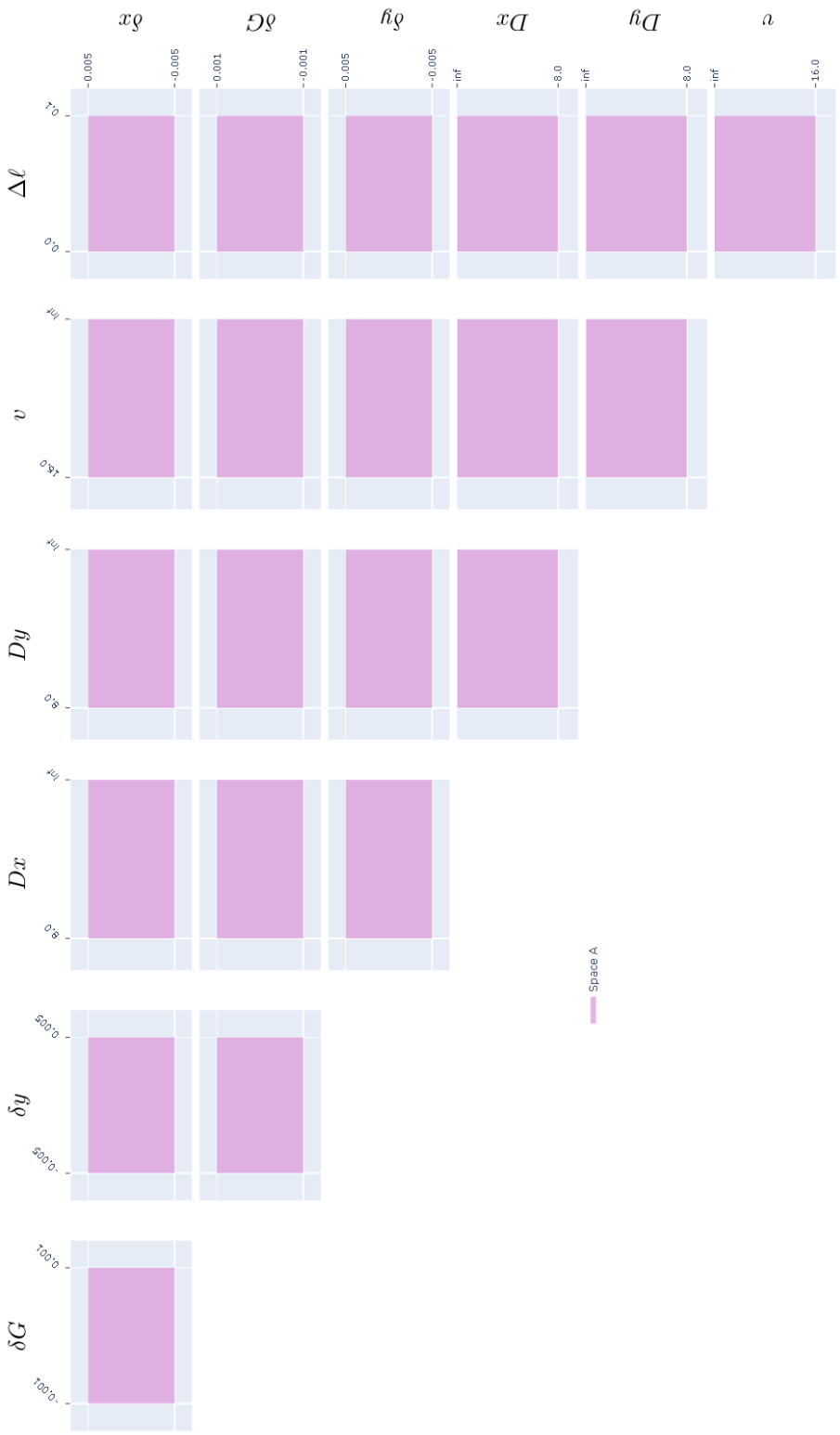

Figure 3.1: Problem space for the example case.

for a separate scaling step. Also, it is important that the shape of the curve reflect the unique contributions of the criteria they represent since the value added or lost generally does not scale linearly with changes in physical characteristics or performance. In the case of print resolution, a negative exponential curve is chosen for $u_{\Delta\ell}$, indicating that each unit of improvement in resolution adds more value that the previous one. Similar to the relation between cost and tolerance in traditional manufacturing methods. For print area, the opposite phenomenon is present, where each increase in print area adds less value than the previous one. This is because as the printable area becomes larger, it increases the size of parts that can be printed but quickly eventually it becomes impractical to print over such a large area due to challenges such part cooling, print times, and frame stiffness. The area itself does not directly cause these problems, but the value is lessened as a result. Finally, the curve for travel-speed most highly rewards velocities in the vicinity of 28 in/s. It has a more gradual slope above this point than below it, reflecting that erring on the high side is preferable to being slower. The reason higher velocities decrease in utility is that it necessitates high acceleration to achieve in a confined space and entails high momentum, both of which increase stress on other components. These drawbacks could also be addressed by adding more criteria to account for those issues separately, allowing a simpler utility curve for velocity. In the interest of presentation, though, applying a single curve reduces the dimensionality of the plot and helps to illustrate that curves can be as simple or complex as necessary to capture the intended relationship.

These utility curves are then weighted and combined to form the response surface that communicates the overall utility of any solution (or non-solution design) being considered for the project. The utility response U for this problem is a linear combination of the curves scaled by their assigned weight. The weights used here sum to 1, which preserves the 0-1 scale from the individual curves.

$$U = w_{\Delta\ell}u_{\Delta\ell} + w_a u_a + w_v u_v \tag{3.7}$$

with $w_{\Delta\ell} = 0.5$, $w_a = 0.2$, and $w_v = 0.3$. The resulting surface, depicted in Figure 3.3, is used to assign the utility score to each design point. It provides the color gradient shown in each solution space and quality space plot below.

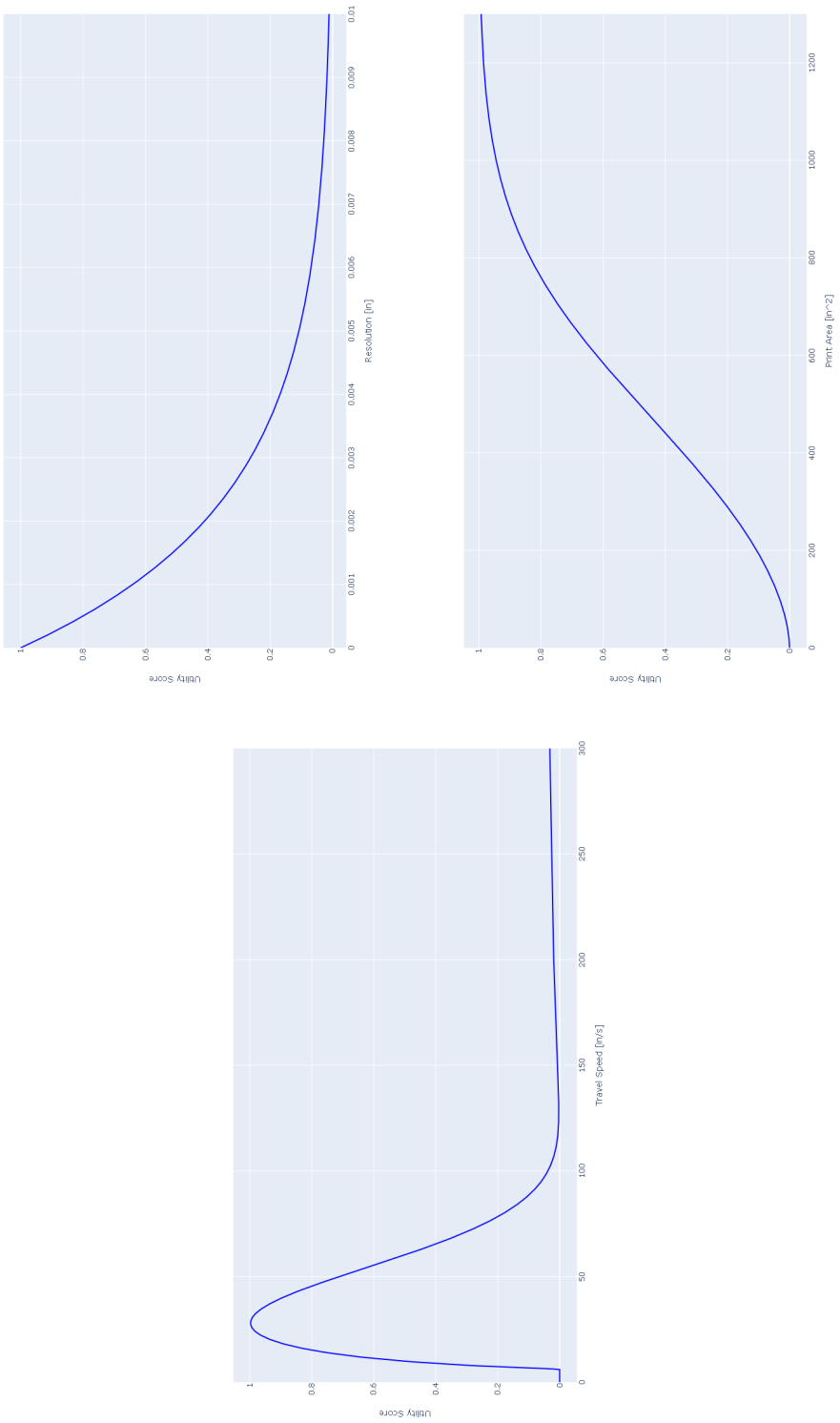

Figure 3.2: Utility curves for the resolution, print area, and travel speed.

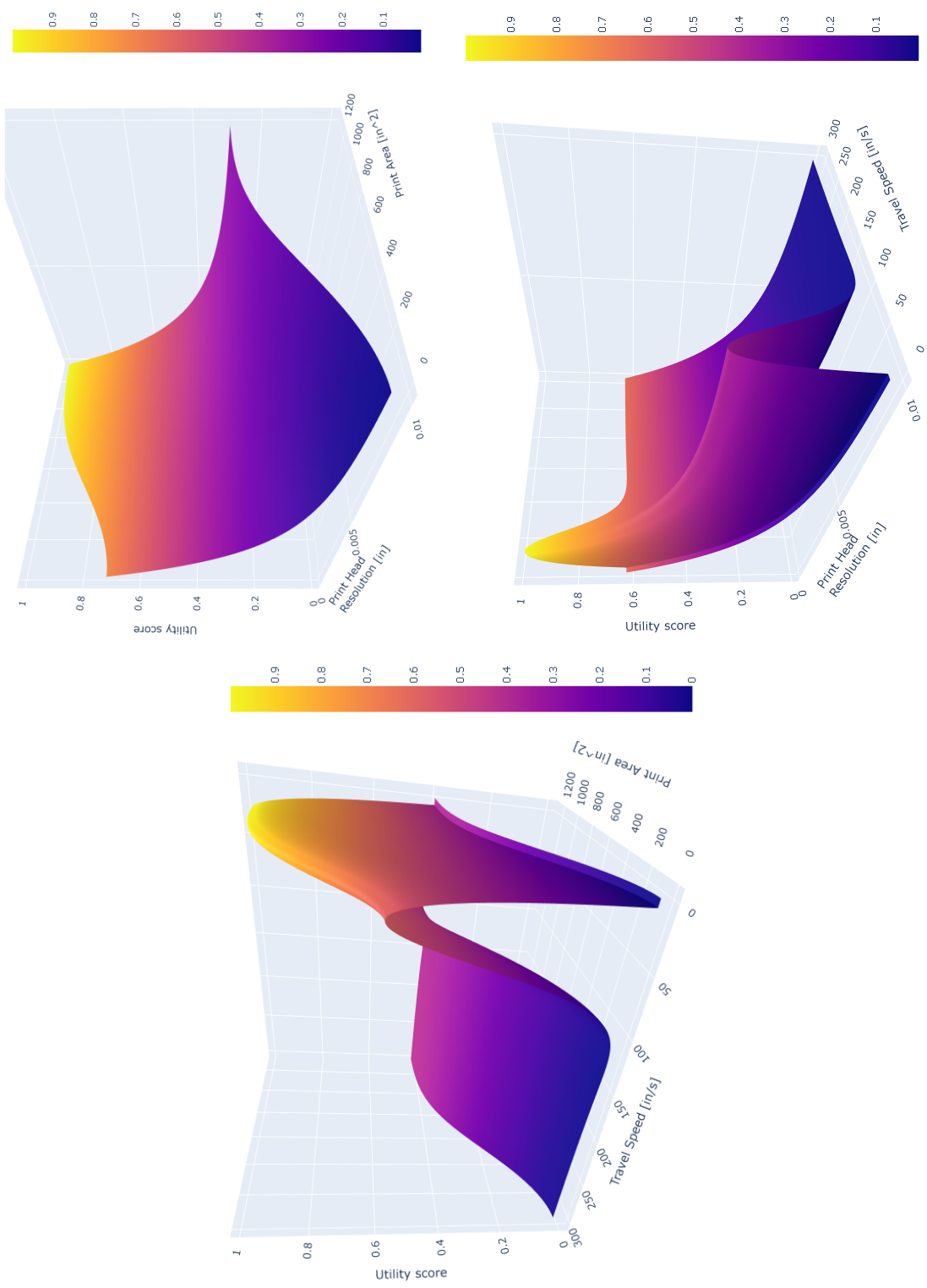

Figure 3.3: Response surface projected onto criteria axes pairwise.

3.2.2 Design Concept A

Concept A, H^A, is characterized by a cuboid frame, with an overhead gantry that carries the print head. The frame is aluminum extrusion in a cube configuration with stiffening plates attached to the back and sides. A belt and pulley system in a CoreXY layout is used to maneuver the print head. This is the format that the original designer of this printer chose to use for the actual build. Figure 3.4 depicts a color-coded 3D rendering of the actual printer design. The colors show the relevant components on which physical constraints have been placed so that the reader may visualize what is being discussed.

To construct the problem map ${}^p\mathcal{M}$, a decision is made for how each requirement is to be met by the design concept. The decisions made in this step will often change throughout the design process and the map will vary correspondingly. For the laboratory 3D printer in this demonstration, one of the goals of the designer was to use existing materials within the lab in order to save money on materials and time in selection. This goal informs many of the design decisions that were made.

Frame Deformation It was decided that the frame would have a skeleton made of T-slotted aluminum extrusion with aluminum plates for reinforcement. The plates are bolted to the frame around three sides. To simplify the model, only the plates are considered in the equations, with the stiffness of the extrusion ignored, and the deformation of the plates is considered as a beam bending problem. This implies two critical assumptions. First, it assumes that the forces on the plates only induce in-plane shear and normal stresses. And second, it assumes that the plates will bend in-plane before shear-buckling occurs. These were considered reasonable given the relatively small forces involved (<3 lbf at max acceleration) and the amount of allowable deflection in the constraints ($\geq$ 0.001 in). This also assumes that all forces are resisted by the plates alone and not the extrusion. Accounting for the load sharing between the plates and the extrusion complicates the model and is likely not necessary. The two together are stiffer than either alone, so if it can be shown that the plates alone are able to meet the requirement then the two together will also. The result is given by Equation 3.8.

$$f_{\delta_x} = \frac{F_{head}\, s^3}{Y\, t_{plate}\, w_{plate}^3} \tag{3.8}$$

Where F_{head} is the force caused by the acceleration of the print head in the x-direction, s is the

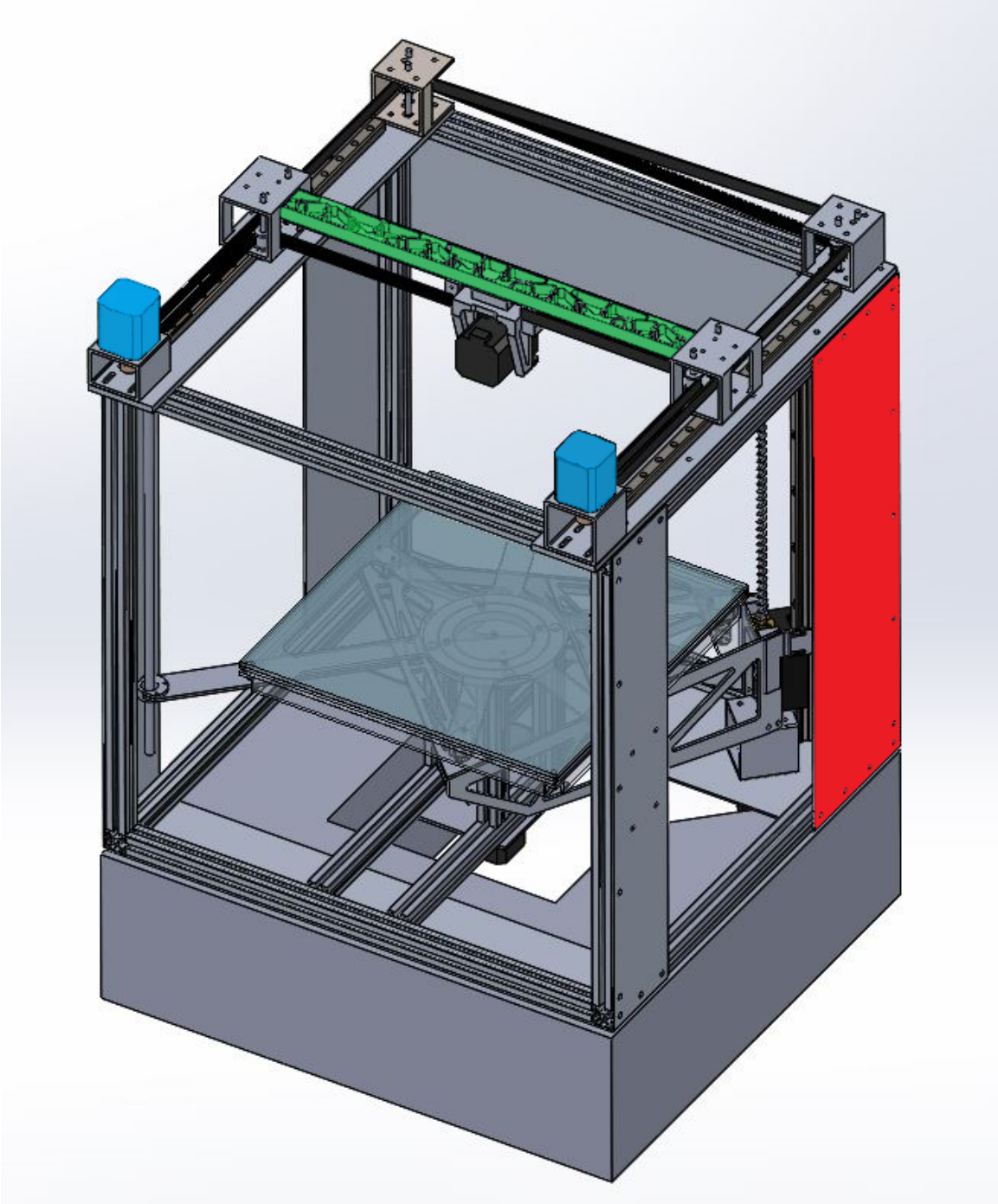

Figure 3.4: The printer that the example case is based on. The red area marks the stiffening plate for the frame; green represents the gantry bar; and blue shows the location of the motors (Roberts, 2021).

side length of the frame, Y is the modulus of elasticity of the reinforcing plate material, t_{plate} is the thickness of the reinforcing plate, and w_{plate} is the width of the plate. Note that s, Y, t_{plate}, and w_{plate} are design parameters, while F_{head} is not. That is because the design of the print head is not under consideration in this context. To consider that aspect of the printer design, this term would need to be decomposed into the mass and acceleration components, which in turn would be decomposed recursively until only independent design parameters remained. In the interest of balancing complexity with clarity, the force will be considered a fixed, known quantity here.

The equation for deflection in the y-direction is the same except that the force involved must account for the motion of the gantry along with the print head.

$$f_{\delta_y} = \frac{F_{gantry}\ s^3}{Y\ t_{plate}\ w_{plate}^3} \tag{3.9}$$

The plate reinforcing the y-direction is physically separate from the one for the x-direction. However, their dimensions will be the same, so the same variables are used.

Gantry Bar Deformation The gantry bar is to be constructed of a solid piece of aluminum. To save weight, a pattern has been chosen that will allow a portion of the material to be removed while continuing to provide stiffness. The pattern consists of a series of triangles cut downward into the gantry but not all the way through. The specific dimensions of the bar and pattern are the design variables to be considered for this purpose, which include the space between the patterned triangles t_{wall}, the depth of material removed t_{floor}, the overall width and height of the gantry w_{gantry} and h_{gantry}, respectively, the gantry's length L_{gantry}. Figure 3.5 shows the pattern and the dimensions. The closed-form calculations to compute the deflection in such a structure, even with the simplistic loading model used here, is extremely cumbersome. To determine the range of dimension values that can satisfy the requirements, ANSYS was used to run a structural FEA on a set of predetermined design points (ANSYS, Inc., 2020). Therefore, the set $\mathcal{M}$ for this case study will include the ANSYS model as f_{δ_G}. While it simplifies the process of testing points in the form space, this approach does limit the number of points that can be tested in a reasonable time frame on a standard PC. This limitation is addressed by using a machine learning technique known as an ensemble classifier (Pedregosa et al., 2011). The classifier is trained on the data generated by ANSYS, after which it can accept new input dimensions and predict whether the provided design would meet the constraint. The classifier can therefore be used as the map function for gantry deflection as it will perform the

necessary test on points in the form space.

$$f_{\delta_G} = \text{Ensemble classifier} \tag{3.10}$$

In cases where precision is less of a restriction than time, such as early in the design process or for non-critical elements, a simple interpolation or regression model could also be used. Conversely, as the results become more critical or when fewer points are being considered, it may be more advantageous to use the ANSYS model itself as the map function. Or, for that matter, physical testing and prototyping may also be where time and resources warrant it.

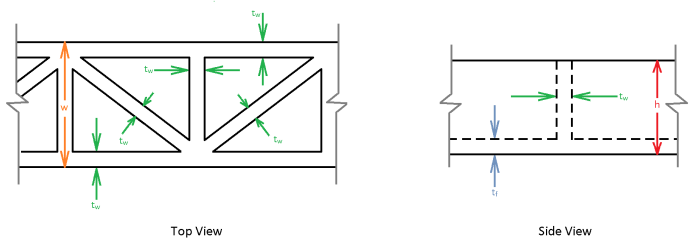

Figure 3.5: Problem space for the example case.

Print Head Travel The travel distance of the print head in each of the lateral directions is a function of the dimensions of the print head in the x- and y-directions as well as the frame and rail support plates. Equations 3.11 and 3.12 show how these are calculated.

$$f_{D_x} = L_{gantry} - w_{head_x} \tag{3.11}$$

$$f_{D_y} = s - w_{head_y} - 2d \tag{3.12}$$

Where w_{head} is the dimension of the head in each of the respective dimensions and d is the cross-sectional dimension of the extrusion used in the frame.

Travel speed The travel speed of the head when not actively printing is calculated at the max motor torque of a given motor. Although stepped motors can travel faster at lower torque values when no load is applied, it is difficult to determine exactly how fast the motor will actually turn when incorporated into a system due to all of the various parasitic losses in play, such as belt stiffness, pulley friction, and friction. Therefore, the speed at max torque is used here to ensure the

fast-travel requirement is met. The velocity of the print head is dependent not only on the motor speed but also the drive components which transmit the rotational velocity ω of the motor shaft to the linear velocity v of the head. In this case, that transmission is accomplished using a belt and pulley system. This provides two equations that must be combined to get the function f_v for the map set (Equations 3.13 and 3.14). The combination gives Equation 3.15 which will be used for the map.

$$\omega = 360\,\frac{LI_{max}}{V\pi} \tag{3.13}$$

$$v = \frac{pN\omega}{\pi} \tag{3.14}$$

$$f_v^A = 360\,\frac{pNLI_{max}}{V\pi^2} \tag{3.15}$$

Where p is the circular pitch of the pulley, N is the number of teeth of the pulley, L is the motor inductance, I_{max} is the maximum rated current of the motor, and V is the rated motor voltage. The superscript 'A' here indicates that this equation is associated only with design A. It is necessary to distinguish this from another mapping function used for concept B to calculate the same constraint parameter value.

Print resolution As with the travel speed, the resolution of the print head is simultaneously dependent on the motor characteristics and the geometry of the pulley. The number of magnetic poles in the motor affects the smallest angular adjustment that the motor can make. The motor controller also plays a role in this as modern controllers usually offer "microstepping" capabilities to improve resolution. Doing so results in a trade-off with stall torque and would normally need to be accounted for but that will be neglected for the purposes of this study. These microsteps μ are generally fractions of the step angle ϕ, the angular distance between poles of the motor. From there, each incremental movement of the motor shaft is related to the movement of the print head via the pulley dimensions. The equation below describes these relationships.

$$f_{\Delta\ell}^A = \frac{\mu pN\phi}{360} \tag{3.16}$$

Problem Map Equations 3.8 - 3.16 are gathered together as the elements of ${}^{p}\mathcal{M}$ and their inputs are similarly collected to form the set $\mathcal{B}$ of design variables as shown below. If $F_{head}, F_{gantry}, w_{head_x}, w_{head_y}$, and d are known and L, I_{max}, V, and ϕ are intrinsically linked according to the motor selection, then

$$ {}^{p}\mathcal{M}^{A} = \left\{ \begin{array}{c} f_{\delta_x}(s, Y, t_{plate}, w_{plate}) \\ f_{\delta_y}(s, Y, t_{plate}, w_{plate}) \\ f_{\delta_G}(L_{gantry}, w_{gantry}, h_{gantry}, t_{floor}, t_{wall}) \\ f_{D_x}(L_{gantry}) \\ f_{D_y}(s) \\ f_{v}^{A}(p, N, motor) \\ f_{\Delta\ell}^{A}(\mu, p, N, motor) \end{array} \right\} \tag{3.17} $$

$$ \mathcal{B}^{A} = \{s, Y, t_{plate}, w_{plate}, L_{gantry}, w_{gantry}, h_{gantry}, t_{floor}, t_{wall}, \mu, p, N, motor\}. \tag{3.18} $$

The cardinality m of the variable set gives the dimensionality of the solution space. In this case, $m = 13$ as that is the number of unique variables needed to calculate a value for each requirement in $\mathcal{R}$ based on the current design concept.

To build the solution space for this case, a sample-and-test method very similar to the statistical sampling in a Monte Carlo simulation is used. A sample space must therefore be defined from which points can be drawn. Each axis of the form space is theoretically unbounded at the upper end, with the exception of the *motor* axis. However, as is often the case in engineering, a reasonable search space can be chosen that envelopes all the values one would expect to use in the design. The selected space can be altered and resampled if a wider search becomes necessary. In the case of this design concept, the search bounds for dimensional variables have been refined to their current bounds based on trial and error. The motor and pulley options were selected based on commonly available stepper motor characteristics. These sampling options are listed below for each design variable. A uniform distribution is assumed in each case.

s: Frame side length

type: continuous
min: 21.0 in
max: 27.0 in

q: Stiffening plate width

type: continuous
min: 2.0 in
max: 10.0 in

t: Stiffening plate thickness

type: continuous
min: 0.10 in
max: 0.20 in

Y: Young's modulus of stiffening plate material

type: continuous
min: 9×10^6 psi
max: 12×10^6 psi

G: Unsupported length of gantry bar

type: continuous
min: 10.0 in
max: 35.0 in

w: Width of gantry bar

type: continuous
min: 0.25 in
max: 1.50 in

h: Height of gantry bar

type: continuous
min: 0.05 in
max: 0.75 in

$\mathbf{t_f}$: Floor thickness of gantry bar

type: continuous
min: 0.01 in
max: 0.20 in

$\mathbf{t_w}$: Wall thickness of gantry bar

type: continuous
min: 0.01 in
max: 0.15 in

μ: Smallest microstepping capability of motor drivers

type: explicit
options: [0.0625, 0.125, 0.250]

p: Pitch of belt pulleys in coreXY system

type: continuous

min: 1.0 mm

max: 3.0 mm

n: Number teeth on pulleys in coreXY system

type: integer

min: 15

max: 50

motor: Stepper motor selection

type: coupled

options:

1: {L: 4.5 mH, V: 3.5 V, Amp: 1.0 A, φ: 1.8}
2: {L: 3.2 mH, V: 4.0 V, Amp: 1.5 A, φ: 1.8}
3: {L: 6.6 mH, V: 12.0 V, Amp: 3.6 A, φ: 1.8}
4: {L: 2.6 mH, V: 6.0 V, Amp: 1.0 A, φ: 1.8}
5: {L: 4.4 mH, V: 24.0 V, Amp: 0.3 A, φ: 1.8}
6: {L: 5.4 mH, V: 6.0 V, Amp: 1.3 A, φ: 1.8}
7: {L: 8.2 mH, V: 12.0 V, Amp: 0.7 A, φ: 1.8}
8: {L: 2.9 mH, V: 3.4 V, Amp: 2.8 A, φ: 1.8}
9: {L: 9.0 mH, V: 12.0 V, Amp: 0.8 A, φ: 1.8}
10: {L: 3.8 mH, V: 3.06 V, Amp: 10.7 A, φ: 1.8}

Note that the motor constitutes a single axis in the form space, with the motor characteristics of a given design being tied to its coordinate value along that axis. Units are listed according to the commonly available options, hence the inconsistency in the system of measurement.

The quality map is constructed in much the same way as the problem map, except that the inputs are known beforehand and the equations that output the criteria values must each take a subset of $\mathcal{B}^A$ as its input. The equations for concept A are

$$f^A_{\Delta\ell} = 3.94\text{e }{-2}\left(\frac{\mu\ p\ n\ \phi}{360}\right) \tag{3.19}$$

$$f_a = (L_{gantry} - 2.5)(s - 6.04) \tag{3.20}$$

$$f^A_v = 5.47\text{e }{-5}\left(\frac{p\ n\ \phi\ V}{Amp\ L}\right) \tag{3.21}$$

$$ {}^{q}\mathcal{M}^{A} = \left\{ \begin{array}{c} f^{A}_{\Delta\ell}(\mu, p, n, motor) \\ f_{a}(L_{gantry}, s) \\ f^{A}_{v}(p, n, motor) \end{array} \right\}. \tag{3.22} $$

which gives the quality map shown in Figure 3.6.

Figure 3.7 shows the solutions space for concept A resulting from the above-provided conditions. The color assigned to the markers indicates the utility value associated with that point. Notice that the gradient in many of the subplots is quite noisy. This is because the axes in those plots do not correlate closely with the utility response surface. The most obvious exceptions to this are the plots for μ. This makes sense owing to the fact that smaller microstepping can drastically increase resolution. The max utility for any design in this solution space is 0.854, yet the mean is significantly lower at 0.382. Since the total range of values for the response surface is guaranteed to extend from 0 to 1, this low average utility may indicate that are ways to improve the concept.

3.2.3 Design Concept B

A second design concept is also included to allow for demonstrating the comparison of dissimilar concepts and some of the ways that changes propagate through the spaces. Concept B H^B will share some of its design variables with H^A, but will also use a few others and will have different maps. This time, instead of a belt and pulley system for X and Y movement, the idea of using lead screws will be explored to see how it compares.

The problem map for concept B shares the same functions for δ_x, δ_y, δ_G, D_x, and D_y but replaces the equations for travel speed and resolution with the following:

$$ f^{B}_{v} = \frac{sps\ p\ \phi}{360} \tag{3.23} $$

$$ f^{B}_{\Delta\ell} = \frac{\mu\ p\ \phi}{360} \tag{3.24} $$

giving a slightly different problem map

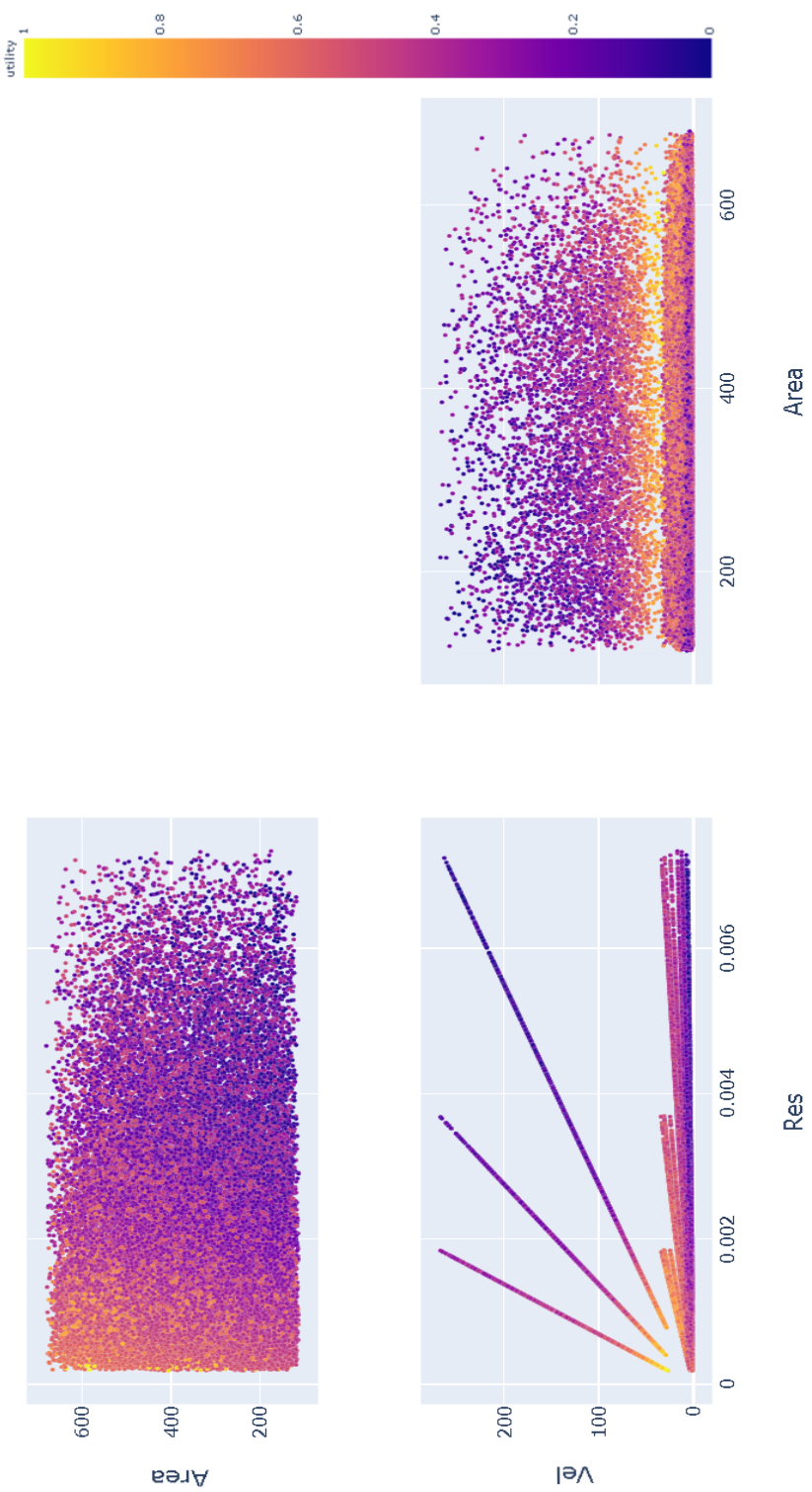

Figure 3.6: Quality space for design concept A with overall utility as the color gradient.

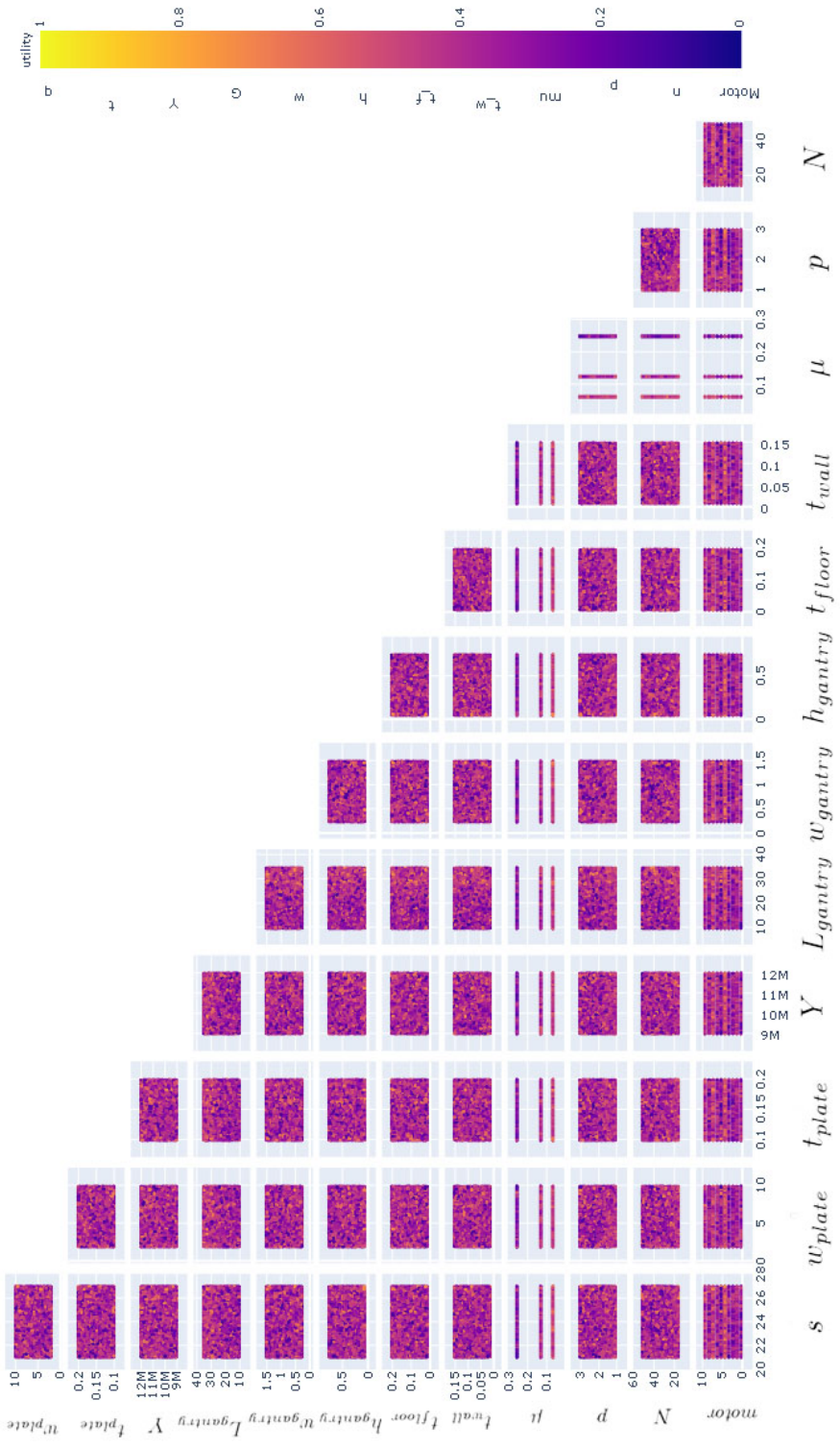

Figure 3.7: Solution space for design concept A with overall utility as the color gradient.

$$
{}^{p}\mathcal{M}^{B} = \left\{ \begin{array}{c} f_{\delta_x}(s, Y, t_{plate}, w_{plate}) \\ f_{\delta_y}(s, Y, t_{plate}, w_{plate}) \\ f_{\delta_G}(L_{gantry}, w_{gantry}, h_{gantry}, t_{floor}, t_{wall}) \\ f_{D_x}(s) \\ f_{D_y}(s) \\ f_{v}^{B}(sps, p, motor) \\ f_{\Delta\ell}^{B}(\mu, p, motor) \end{array} \right\} \tag{3.25}
$$

with the following set of design variables:

$$
\mathcal{B}^{B}\{s, Y, t_{plate}, w_{plate}, L_{gantry}, w_{gantry}, h_{gantry}, t_{floor}, t_{wall}, \mu, screw, motor\}. \tag{3.26}
$$

The variable *sps* in f_v^B is a motor characteristic called "steps per second" and refers to the maximum number of times the motor is able to advance one step in a single second.

The search space of H^B is the same as A for variables s, q, t, Y, G, w, h, t_f, t_w, and μ. However, due to the completely redesigned motion system and the altered problem map, the motor options are different, the pulley characteristics are no longer applicable, and a new component – the lead screw – must be taken into account. The lead screw and updated motor options are listed below.

screw: Lead screw selection

type: coupled

options:

1: {d: 8.0, p: 0.012, η: 21}
2: {d: 8.0, p: 0.049, η: 89}
3: {d: 8.0, p: 0.159, η: 86}
4: {d: 9.5, p: 0.025, η: 21}
5: {d: 9.5, p: 0.125, η: 84}
6: {d: 9.5, p: 0.200, η: 84}
7: {d: 10.0, p: 0.039, η: 79}
8: {d: 10.0, p: 0.196, η: 85}
9: {d: 12.0, p: 0.049, η: 86}
10: {d: 12.0, p: 0.197, η: 88}

motor: Stepper motor selection

type: coupled

options:

1: {L: 3.0 mH, V: 24.0 V, Amp: 1.6 A, φ: 1.8, sps: 10000}
2: {L: 7.7 mH, V: 24.0 V, Amp: 1.0 A, φ: 1.8, sps: 8000}
3: {L: 7.1 mH, V: 24.0 V, Amp: 1.7 A, φ: 1.8, sps: 6000}
4: {L: 2.5 mH, V: 24.0 V, Amp: 2.7 A, φ: 1.8, sps: 8000}
5: {L: 5.8 mH, V: 24.0 V, Amp: 2.5 A, φ: 1.8, sps: 6000}
6: {L: 1.6 mH, V: 24.0 V, Amp: 3.9 A, φ: 1.8, sps: 6000}
7: {L: 3.0 mH, V: 36.0 V, Amp: 1.6 A, φ: 1.8, sps: 10000}
8: {L: 7.7 mH, V: 36.0 V, Amp: 1.0 A, φ: 1.8, sps: 10000}
9: {L: 7.1 mH, V: 36.0 V, Amp: 1.7 A, φ: 1.8, sps: 8000}
10: {L: 2.5 mH, V: 36.0 V, Amp: 2.7 A, φ: 1.8, sps: 10000}
11: {L: 5.8 mH, V: 36.0 V, Amp: 2.5 A, φ: 1.8, sps: 8000}
12: {L: 1.6 mH, V: 36.0 V, Amp: 3.9 A, φ: 1.8, sps: 8000}

Here, d in the lead screw options denotes the diameter of the lead screw, p now refers to the thread pitch of the screw, and η is the power transmission efficiency of the screw.

Unfortunately, the search for points in $\mathcal{S}^B$ under the current problem space yields no results. To troubleshoot, a table is constructed to compare the minimum and maximum values for the tested points against the allowable ranges for each constraint. This should indicate where the problem space and the image of search space under ${}^p\mathcal{M}^B$ are failing to intersect. Although it is possible that the issue is more subtle than simply failing to overlap, this provides a good starting point. Table 3.1 shows the results. The gantry deflection δ_G is omitted from the table since f_{δ_G}, being a binary classifier, only outputs `True` or `False`. However, the set of failed design points contains several `True` values, indicating an intersection with the problem space along the δ_G axis in $\mathcal{C}$. The bolded row shows the axis along which the image and the problem space fail to overlap with each other. The maximum v value from the map is less than the minimum allowable value under $\mathcal{R}_v$. The lead screw concept with the current motor-screw options does not produce high enough velocities to meet the constraint.

There are several ways to address this issue in order to develop a solution space for the concept, but the easiest thing to do is to reevaluate the constraint on print head velocity and determine if the lower velocity is in fact acceptable. This does not necessarily mean that the project must accept the change, but it may be worth the time to see what solutions H^B offers with this constraint relaxed. Then, if the performance of the solutions warrants it, the other ramifications of accepting the change can be evaluated.

Table 3.1: Analysis of the overlap between the mapped points for Concept B and the problem space.

	$\mathcal{R}_1$		${}^{p}\mathcal{M}$	
	min	max	min	max
δ_x	-0.005	0.005	2.6×10^{-6}	1.5×10^{-3}
δ_y	-0.005	0.005	8.1×10^{-6}	4.8×10^{-3}
Dx	8.0	Inf	7.5	32.5
Dy	8.0	Inf	15.0	20.0
$\boldsymbol{v}$	**16.0**	**Inf**	**0.36**	**10.0**
res	0.0	0.001	3.8×10^{-6}	2.5×10^{-4}

So, let a new problem space be constructed based on the old but with

$$\mathcal{R}_2 = \{\mathcal{R}_{\delta_x}, \mathcal{R}_{\delta_y}, \mathcal{R}_{\delta_G}, \mathcal{R}_{D_x}, \mathcal{R}_{D_y}, \mathcal{R}_{v,2}, \mathcal{R}_{\Delta\ell}\} \tag{3.27}$$

where the velocity constraint $\mathcal{R}_v$ is expanded at the low end to 8 in/s.

$$\mathcal{R}_v = \{x \mid x \geq 8\ in/s\} \tag{3.28}$$

Making this change results in a solution space for concept B shown in Figure 3.8.

Now that a solution space has been established for concept B, a quality space may also be found in order to evaluate the resulting solutions. The quality map for concept B will be described by the following equations:

$$f^B_{\Delta\ell} = \frac{\mu\ p\ \phi}{360} \tag{3.29}$$

$$f_a = (L_{gantry} - 2.5)(s - 6.04) \tag{3.30}$$

$$f^B_v = \frac{sps\ p\ phi}{360} \tag{3.31}$$

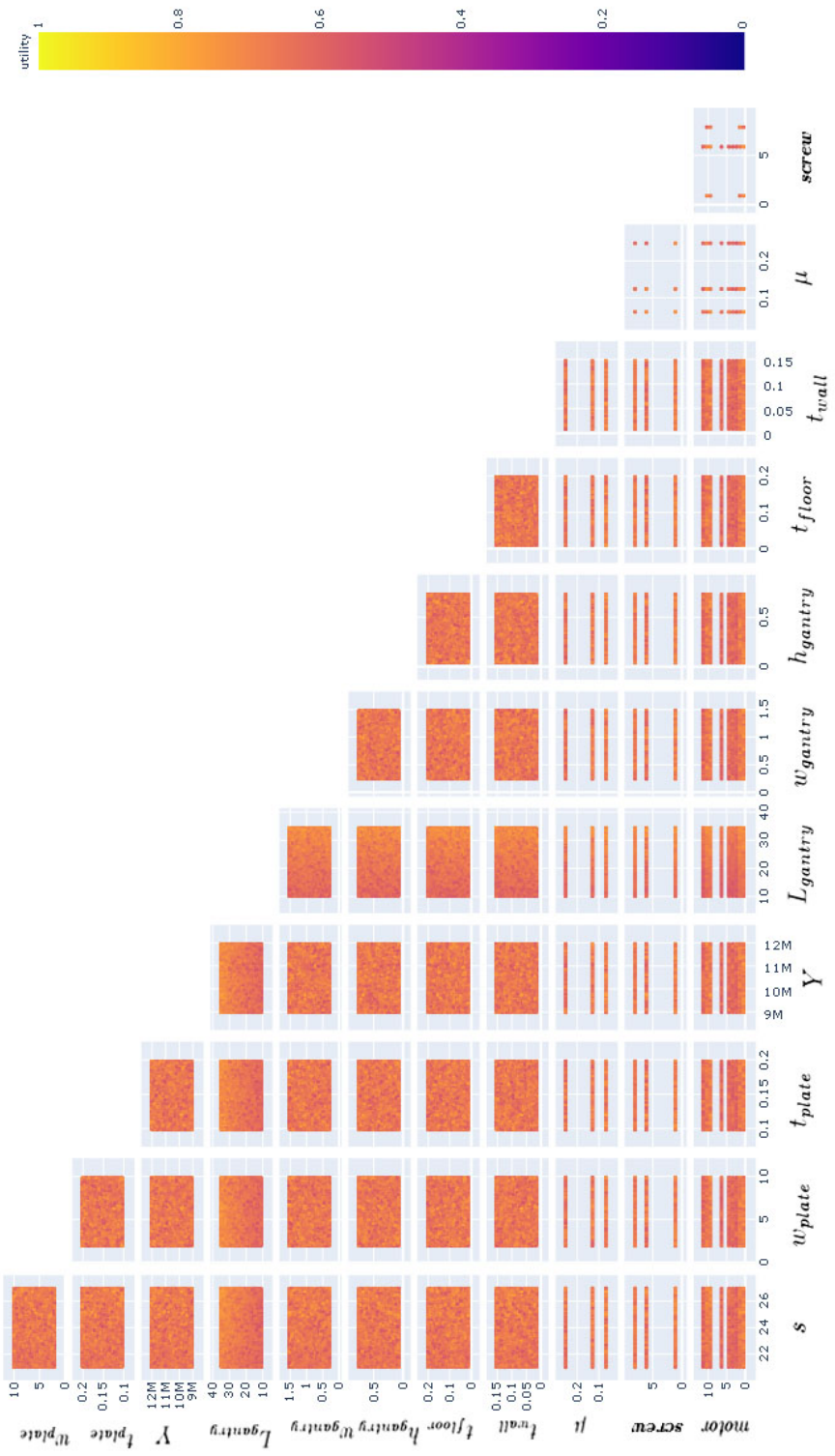

Figure 3.8: Solution space for design concept B with overall utility as the color gradient.

$$ {}^{q}\mathcal{M}^{B} = \left\{ \begin{array}{c} f^{B}_{\Delta\ell}(\mu, screw, motor) \\ f_{a}(L_{gantry}, s) \\ f^{B}_{v}(screw, motor) \end{array} \right\}. \tag{3.32} $$

This map provides the quality space in Figure 3.9. Note that both velocity and resolution are highly discretized since now all design variables for both of these axes are discrete under the new concept.

3.3 Effects of the Change

3.3.1 Effects on Problem Space

Performing a similarity evaluation of the problem space poses a unique challenge for calculating the similarity after a change occurs. In Figure 3.10, the spaces contain infinite boundaries which complicate the ratio calculation. As alluded to in Chapter 2, one method for combating this is to choose a realistic yet sufficiently large upper bound. This can be useful provided the same upper bound is used for the spaces being compared and it makes the problem mathematically tractable. The downside, though, is that if the boundary is too distant then the results may be unhelpful (i.e. arbitrarily close to 0 or 1), even if finite. An alternate approach may be to create a new constraint that provides a useful and potentially dynamic upper bound. This method also demonstrates one way in which this formalism contributes to the evolution of the problem space. In the case of the problem space shown, both methods may be employed to eliminate the infinities. For one, the designer may choose to specify a constrain that the side length must be less than 36 inches to ensure that the printer will fit on a large table. Then the upper bounds for Dx and Dy may use the side length as an upper bound, removing the infinite end-point along the corresponding axes of the space. The upper velocity bound can be set to 300 in/s since, as Figure 3.7 showed, points in $\mathcal{S}_1^A$ come close to but do not reach that speed, and H^B has significantly lower speeds. Therefore, no current solutions would be lost. Figure 3.11 shows the same problem spaces as Figure 3.11 but with finite bounds implemented.

Having made these changes, the spatial similarity is calculated to be 0.986. The two spaces are highly similar since the only difference between them is the change to the lower bound of the

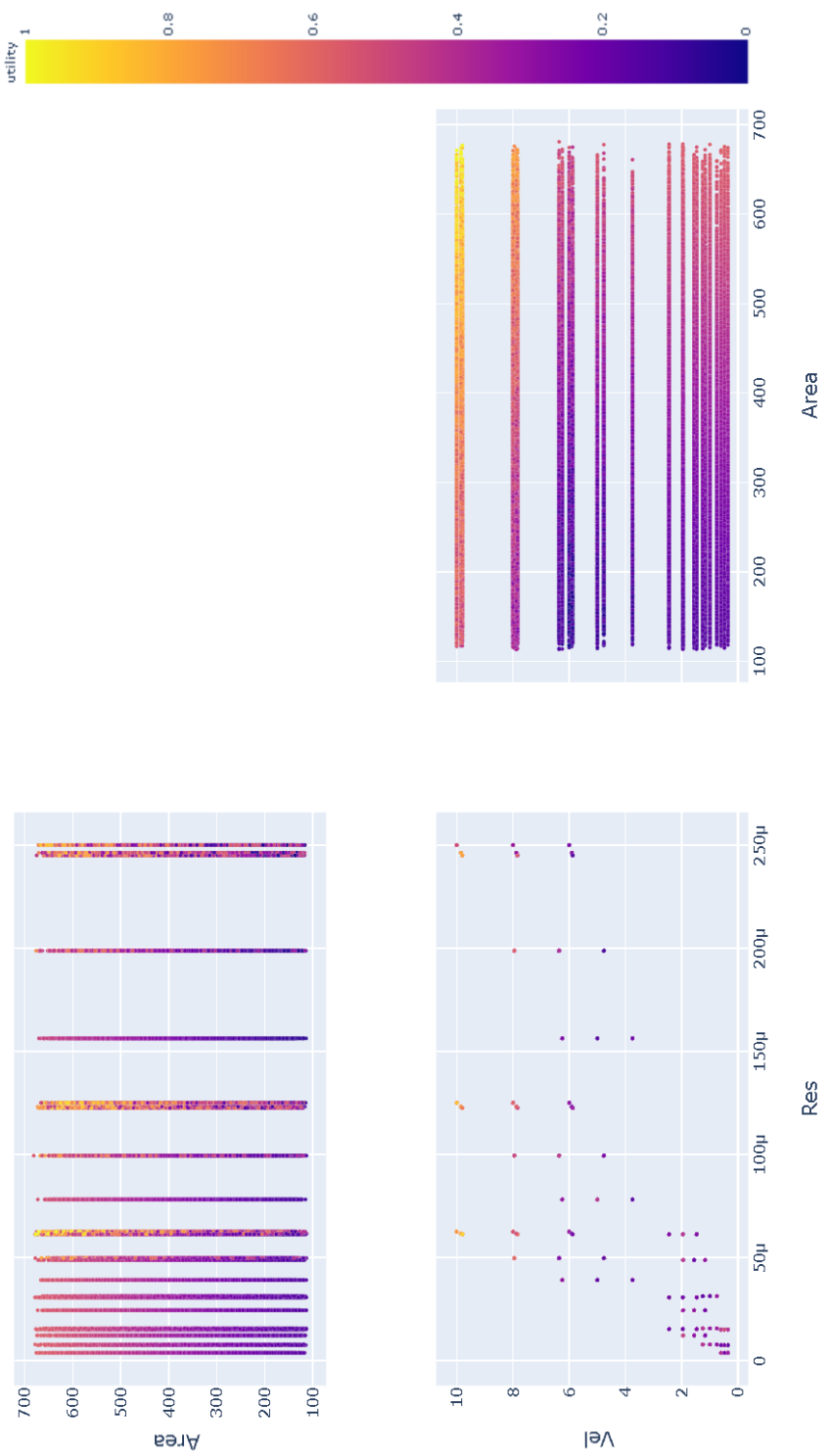

Figure 3.9: Quality space for design concept B with overall utility as the color gradient.

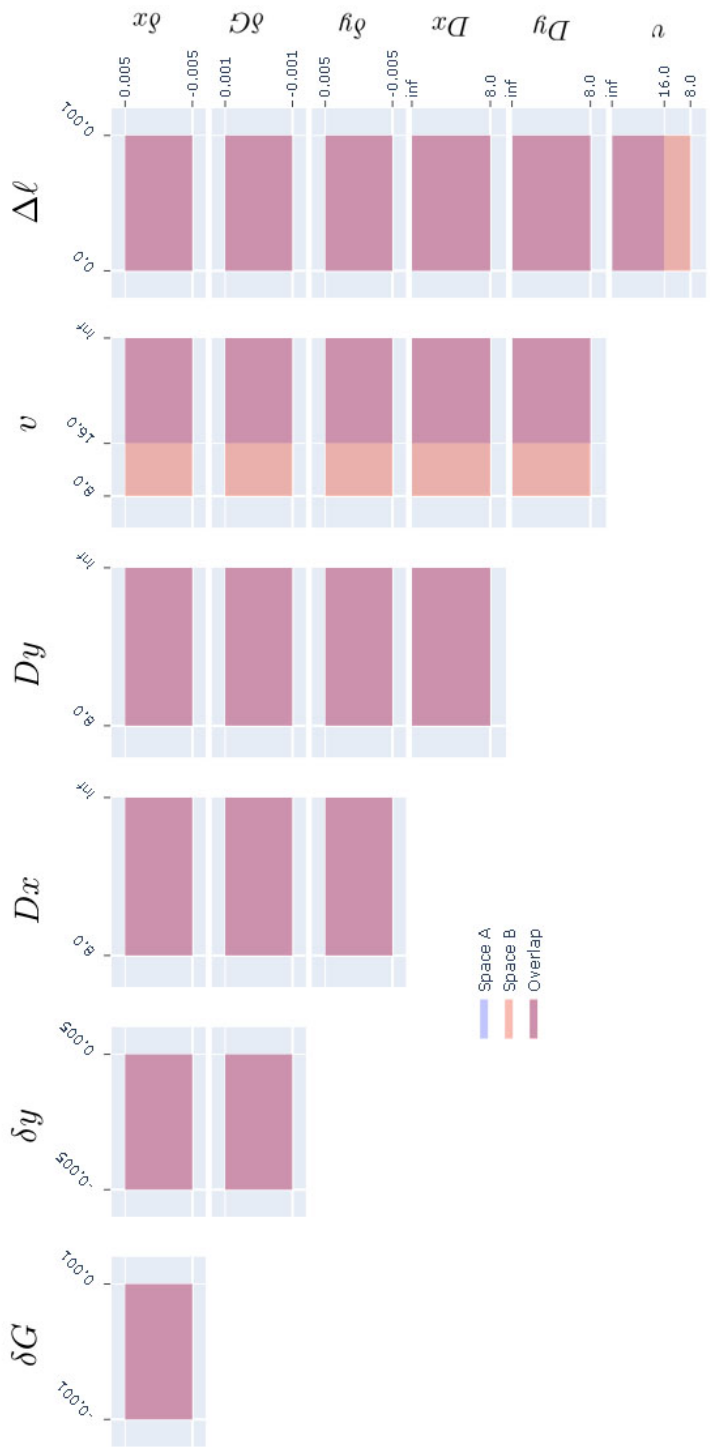

Figure 3.10: Overlay of the problem space before ($\mathcal{P}_1$) versus after ($\mathcal{P}_2$) the adjustment of requirements.

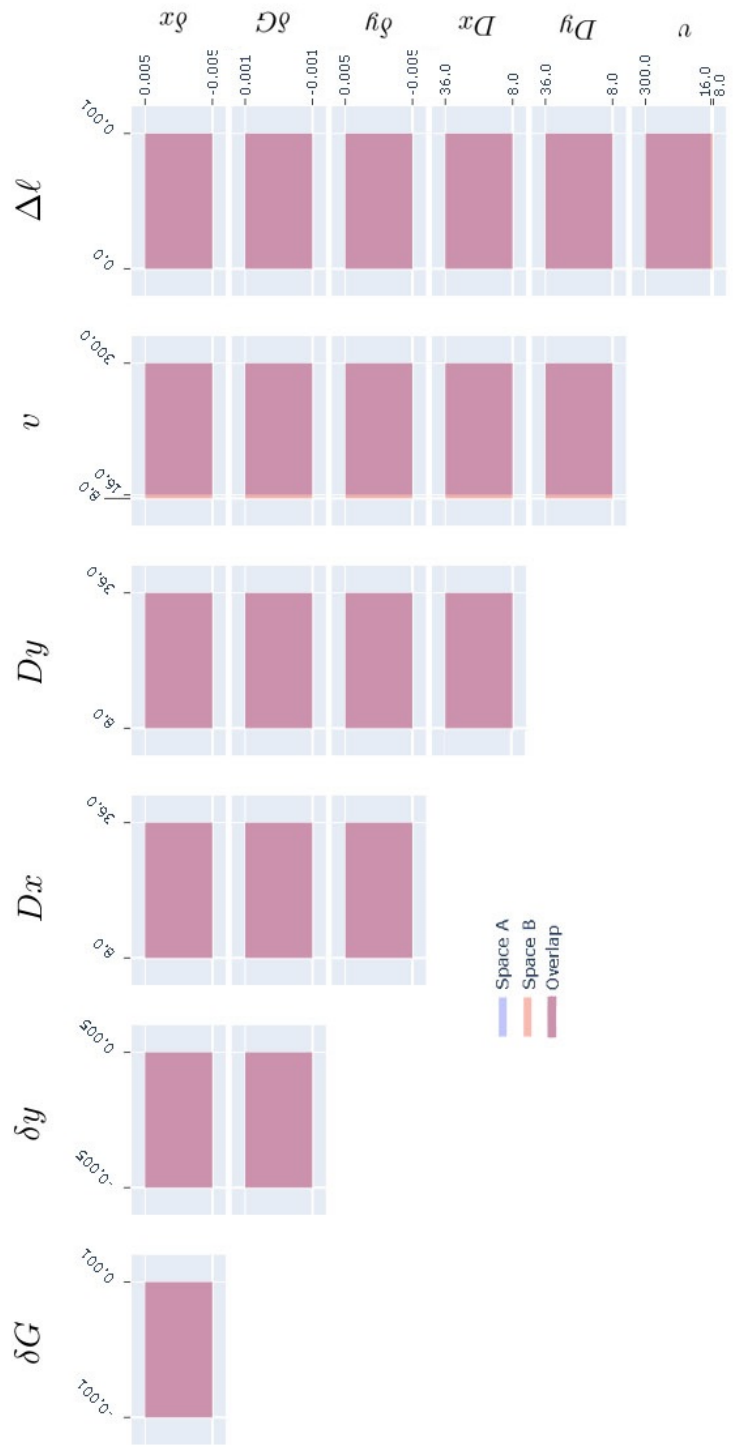

Figure 3.11: Overlay of the problem space before ($\mathcal{P}_1$) versus after ($\mathcal{P}_2$) the adjustment of requirements, with finite bounds.

$\mathcal{R}_v$. Even after reducing the infinite bounds, this difference adds only a relatively small amount of extra volume to the space and so results in a high similarity.

3.3.2 Effects on Concept A

Next, the change to the requirement set needs to be propagated to its dependent spaces, the solution and quality spaces for concept A that were generated previously. Figures 3.12 and 3.13 show the updated results. It is difficult to tell any significant difference from the quality space. However, the difference becomes obvious in the solution space thanks to the noisy colormap. Since the colors are based on coordinates in the objective space, they are grouped with like colors and change somewhat gradually in $\mathcal{O}$. However, points of very different utility scores often end up nearer one another in the form space and so the change in average utility score is more apparent in the blended hue of the applied gradient on $\mathcal{S}^A$. The cause for the higher overall utility is that many of the design points that previously failed to meet $\mathcal{R}_v$ offered much better resolution than the ones having the higher velocity, and resolution is weighted more heavily than velocity. This is one of the design trade-offs that comes with both belt and screw drive mechanisms. This is further evidenced by the fact that concept B, while far slower in general, also exhibited much finer resolution in almost all cases.

Figure 3.14 shows the overlay of $\mathcal{S}_1^A$ and $\mathcal{S}_2^A$. The calculated similarity between these two spaces is 0.766 on a scale of [0, 1]. This indicates there is a large amount of overlap between the two, which stands to reason since the only change has been to widen the allowable region for the velocity constraint. One would expect that this added a certain number of designs to the solution space without excluding any of the previous ones. The plot supports this, showing points that lie in both $\mathcal{S}_1^A$ and $\mathcal{S}_2^A$ as well as some that lie in $\mathcal{S}_2^A$ alone. Yet there are no points that belong exclusively to $\mathcal{S}_1^A$, indicating that none were lost in the change. Also, the vast majority of the points in the pairplot satisfy both sets of constraints ($\mathcal{R}_1$ & $\mathcal{R}_2$) and a relative minority only satisfies the second set. From this fact, it can be inferred that the effect of this change was to expand the solution space such that the original solution space is completely enveloped by the new space, which happened with the problem space as well. This may or may not be desirable in any given project, depending on the design stage, the context, and the goal in making the change. In any case, this method allows for the effect to be quantified and measured thereby assisting in the decision-making process.

Using the problem space similarity from Section 3.3.1 and the solution space similarity, the

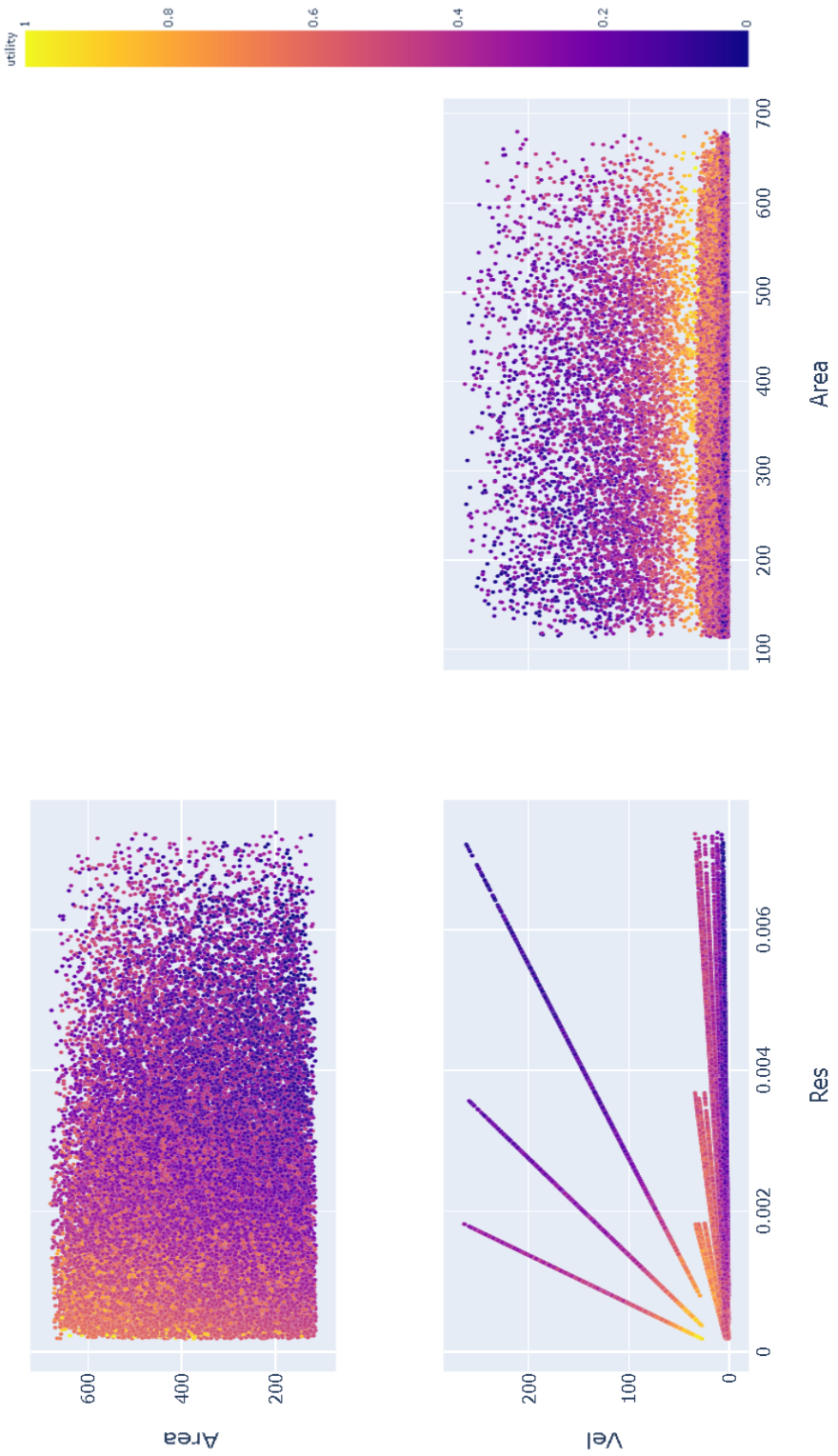

Figure 3.12: Quality space for design concept A following the change to $\mathcal{R}_v$.

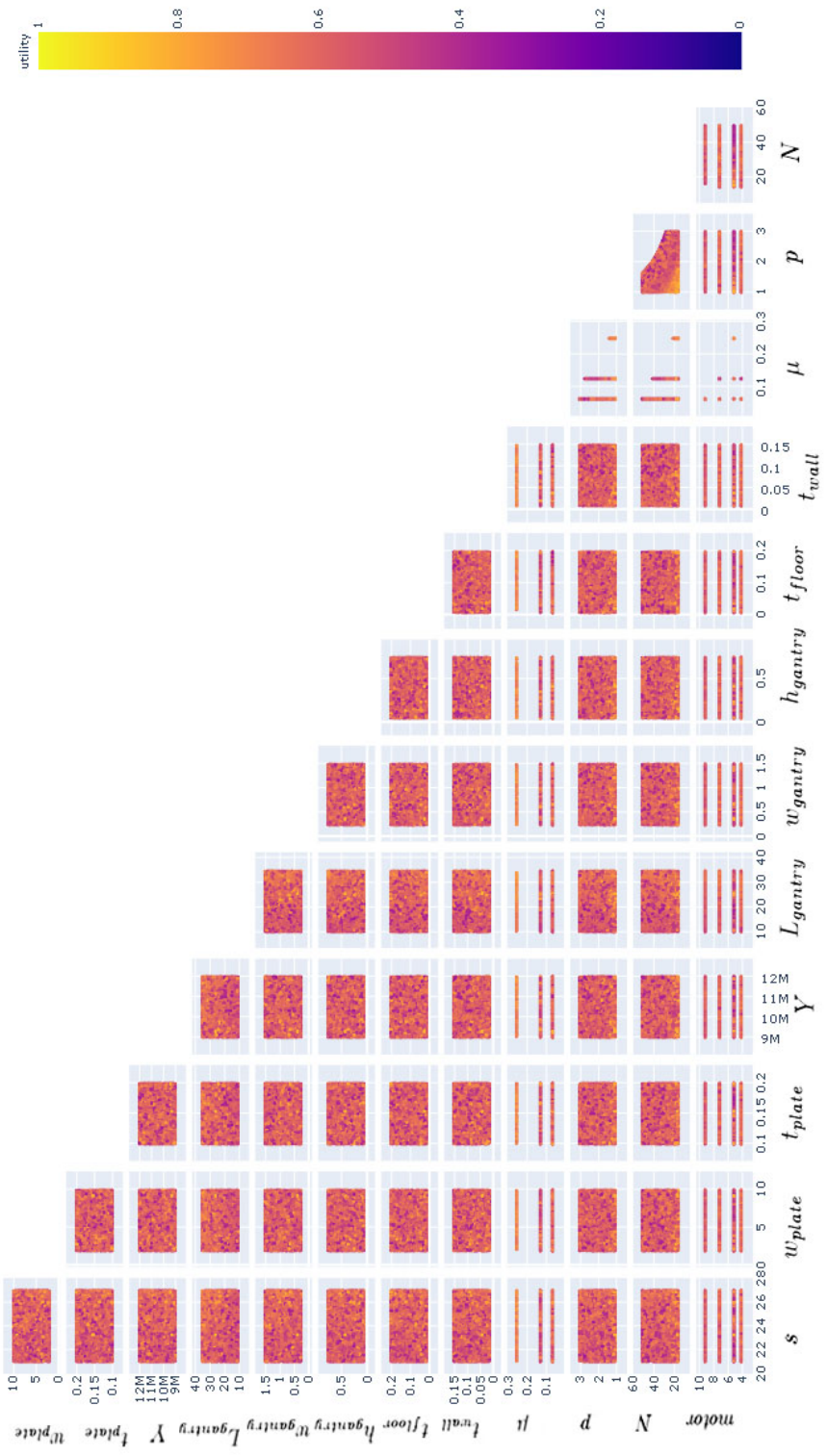

Figure 3.13: Solution space for design concept A following the change to $\mathcal{R}_v$.

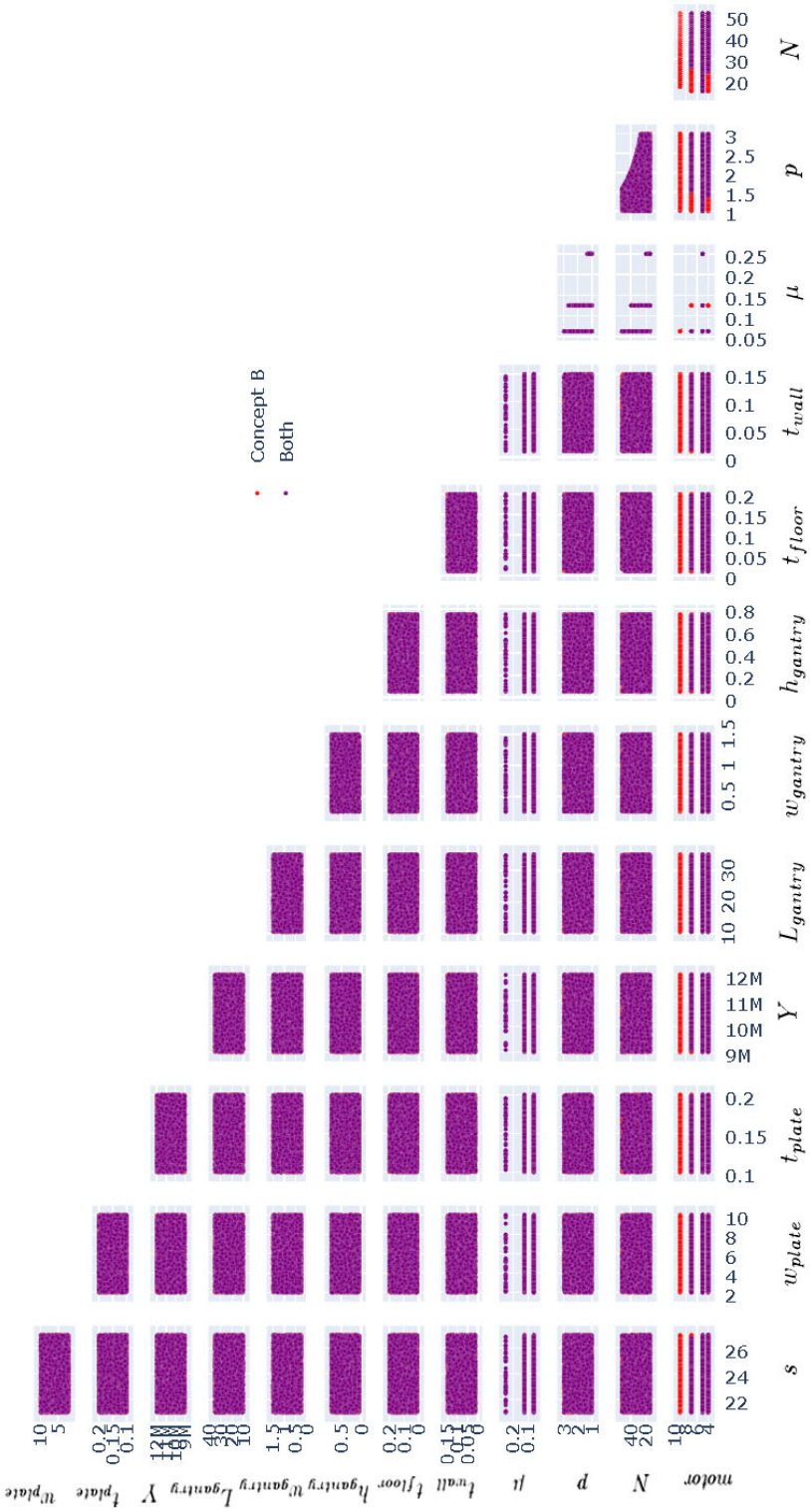

Figure 3.14: Overlay of the problem space before ($\mathcal{S}_1^A$) versus after ($\mathcal{S}_2^A$) the adjustment of requirements. Similarity of the space before and after the change is calculated to be 0.769.

sensitivity of the solution space to the change in the problem space can be calculated as described in Section 2.3.2. In doing so, a sensitivity of 16.71 is obtained. This indicates that the relatively small change to the problem space was amplified in significance as it propagated to the solution space. In this case, it added more volume to the solution space than might have been expected. Given that the range of the sensitivity calculation has no upper bound, the observed effect here may not be exorbitantly large. As with any rate of change, though, the value itself must be considered in the context of the situation.

3.4 Comparing the Concepts

Finally, the two concepts H^A and H^B can now be contrasted against each other. Since they do not share all of the same axes in the form space, the only direct spatial comparison that can be done is between their respective quality spaces, $\mathcal{Q}_2^A$ and $\mathcal{Q}^B$, which can be seen in Figure 3.15. The most noticeable characteristic of this overlay is the large difference in the volume of the two spaces. The number of points in $\mathcal{Q}^B$, on the other hand, is much greater. The figure was generated by randomly sampling 60,000 individual points from each concept's associated search space and testing them against the constraints. Passing points were then mapped into the objective space and their utility values calculated. $\mathcal{Q}^B$ in the figure is composed of more than 8,000 of the tested points while $\mathcal{Q}_2^A$ has only ∼4900 points. Moreover, while packed into a smaller volume, the points of $\mathcal{Q}^B$ exist in a relatively high-value region of $\mathcal{O}$, giving it an average utility of 0.6588 against the average for $\mathcal{Q}_2^A$ of 0.5747. Concept B's utility values also have a much narrower spread and standard deviation, 0.2345 and 0.0460, respectively, versus 0.5415 and 0.0909. That said, Concept A_2 has a significantly higher peak utility value, offering a max utility of 0.8766 whereas Concept B only reaches 0.7788 for its max. The practical implication of this information will likely differ from project to project. But as an early design-stage realization, Concept B may provide more assurance that it will be robust against future changes since it seems to provide a larger number of high-quality design options. Assuming the trade-off in fast-travel speed is deemed acceptable.

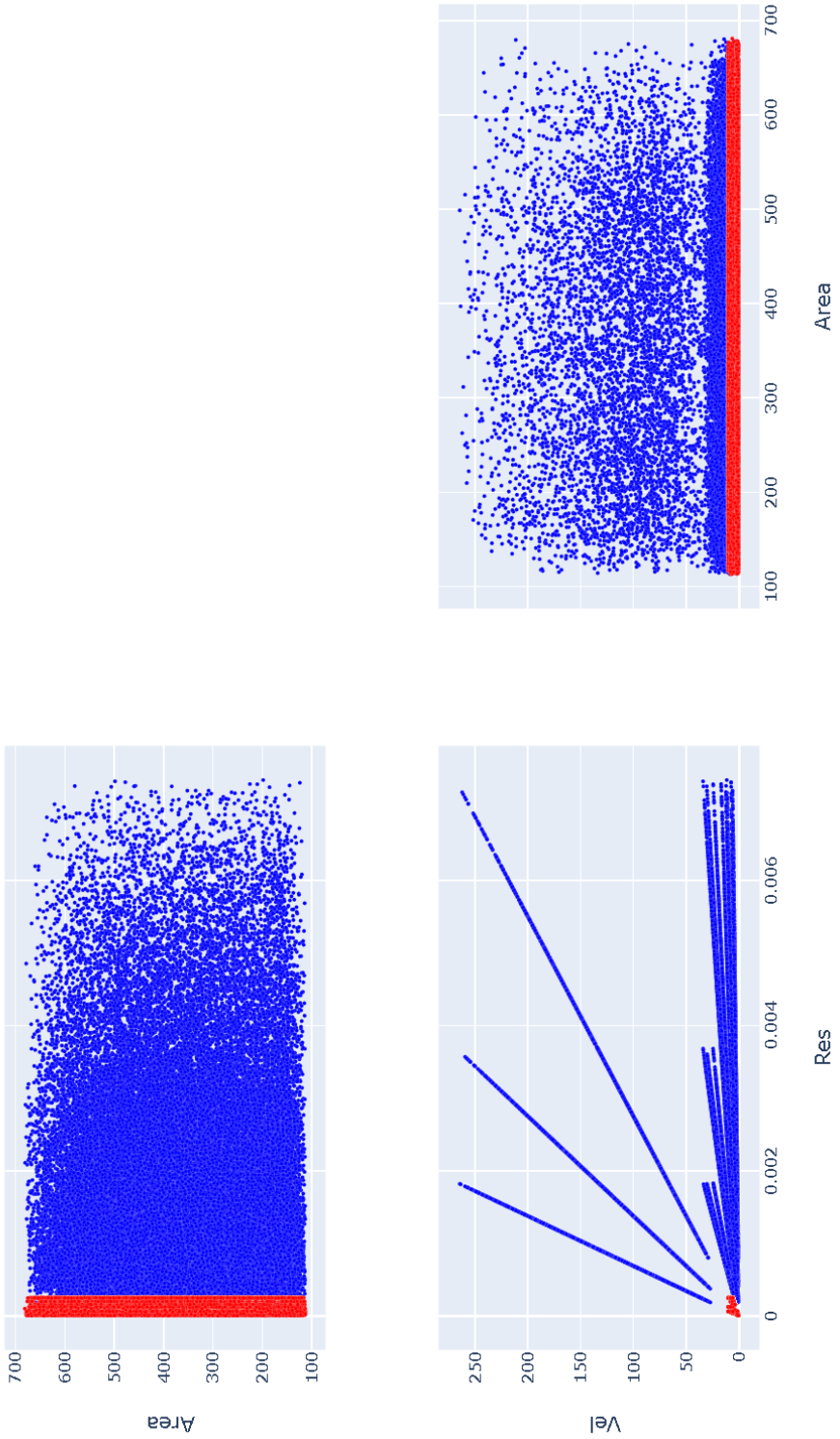

Figure 3.15: Comparison of the quality spaces for both concepts.

Chapter 4

Conclusion

4.1 Discussion

In this thesis, a formal definition has been put forth for three types of design spaces – the problem space, the solution space, and the quality space – along with a problem map and a quality map that establish the mechanism for interaction between the spaces. In addition, a utility response surface has been provided for evaluating the relative performance and merit of design concepts that may have very different traits. Along with these constructs, methods have been proposed for analyzing the change within and between the spaces, and examples were used to demonstrate how the tools are employed and how various complications can be handled in a design project.

The basic motivation for creating these definitions is to provide useful and practical tools that complement techniques engineers already use, such as curves and points in space, while offering new or more streamlined ways to accomplish design tasks. It also seeks to let designers interact with design spaces in a more structured and formal manner. Point-set topology seemed like a natural way to embody the concept of a design space in the context of product engineering while addressing these overarching motivations.

This formalism also contributes to the broader engineering community by complementing or extending some of the existing theories and methods from prior research. In Braha and Reich (2003) the authors discuss the use of topological structures for modeling the design process but intended their work as an exercise in understanding that process rather than enabling it directly. Nonetheless, their model offers a guideline for how the formalism in this thesis might be employed

in a holistic way, whereas the examples in Chapters 2 and 3 only consider brief snapshots. The constraint and form spaces discussed here in Section 2.1 can be considered instances of the function and structure spaces in Braha and Reich (2003), respectively, with the problem and solution spaces being their 'closures'. By convolving the two research efforts, this work supports the authors' stated goal of "developing increasingly better mathematical models of design" and "producing ideas for implementing design support procedures". Both frameworks are based in point-set topology, with Braha and Reich (2003) taking a more strategic approach, so the methods in this paper can add specificity to theirs while remaining compatible with the operations and algorithms they describe.

Another framework where this work may be applied is that of Maher et al. (1996). They discuss the use of genetic algorithms to systematically evolve the problem and solution spaces concurrently. Specific designs and problem parameters are encoded as concatenated binary strings which can then be modified automatically through one of two algorithms and assessed for suitability as a solution. This is highly compatible with the spaces proposed in this paper. Points in the form spaces are ordered pairs that can be readily converted to binary strings and modified according to their methods, then converted back into spatial coordinates and mapped to the problem and quality spaces for evaluation. Maher et al. (1996) go so far as to propose the use of evaluation functions to assess the viability of genetic mutations, a concept already built into the utility response curve from this thesis. Co-evolution offers a powerful means to explore the spaces proposed here, while in return this formalism offers co-evolution a more functional definition of design spaces that can also be used in other ways when exploration is not the primary focus.

Zimmermann and von Hoessle (2013) also provides a gainful tool that can be implemented with this characterization of design spaces. In that paper, an algorithm was described that systematically refines the solution space while ensuring maximal design robustness against uncertainty. Here, only a naive search was demonstrated which may not always be practical or worthwhile. Yet, thanks to the similar representation of the solution space in their paper, it would seem that their method can be smoothly adapted to work with the form space proposed here to improve search time and results.

These points address the goals put forth in Section 1.2, recapitulated here:

1. Has an explicit mathematical defintion

2. Has practical applications to engineering design scenarios

3. Can be extended and improved upon by others in the future

4. Will address gaps identified in current literature

Mathematical Defitinions Section 2.1 laid out a variety of definitions rooted in set-theoretic and topological terms that precisely describe each constituent portion of the framework. The definitions provided allows for these various entities to be used in a manner consistent with other topological formalisms and methods. They also clearly distinguish their respective components from other objects.

Pratical Applications Chapter 3 investigates the use of this new formalism and its methodologies in the context of an existing design project to demonstrate its application in a more pragmatic scenario. This case showed how the various aspects of the framework behave when higher-dimensional spaces are involved and addressed several complications that may arise in real-world implementation.

Extensibility This aim remains to be exhibited since, by definition, its truth relies on the adoption of these methods and tools by others in the research and design communities. However, hopefully, the reader may see that such extension is possible through the efforts delineated here.

Addresses Gaps As discussed above, the current work fills many of the roles left undefined in prior work. The problem and solution spaces from this thesis are synonymous with those discussed in other works and lend themselves to the uses proposed before. The problem and quality maps herein provided expound on the concepts of synthesis and refinement from Maher et al. (1996) and Braha and Reich (2003). Each unit of this framework fits with the design space of Bowen and Dittmar (2017) as an artifact or relationship. And finally, where Zimmermann and von Hoessle (2013) and Maher et al. (1996) discuss the exploration of solution spaces, both the quality and solution spaces from this paper are easily adapted for use.

4.2 Future Work

The formalism and methods that have been proposed here were intentionally designed to be extensible as possible. And while not initially designed to fit within any specific framework found in previous research, a fortunate coincidence of this extensibility seems to be that this proposal may

fit well into the gaps and generalities of certain other formalisms. Confirmation of this will be one of the primary goals of future work in this area. If the spaces and maps depicted here can fill the roles of the spaces postulated in Maher et al. (1996) or Braha and Reich (2003), then it would help to make their work more practical for commercial design applications.

Additionally, the work of Shan and Wang (2004), Graff et al. (2016), and Zimmermann and von Hoessle (2013) provide promising avenues of inquiry as well. Each of these studies discusses methods that are predicated on various topological constructs congruent with those used in this thesis.

Finally, there are several general or 'unified' design theories that have gained wide acceptance, such as the axiomatic design theory put forth by Suh (1998) and C-K design theory developed by Hatchuel and Weil (2009). It would be worthwhile to investigate the potential integration of the methods proposed here into these overarching theories. Such consolidation would help to validate this schema and provide a more complete road map for its interaction with other design tools and techniques.

Aside from integrating this work into other research, there are some specific areas where this formalism can be enhanced and improved as well. One significant extension that would be needed if this is to be used in any full-scale design work is to adapt the spaces and maps for use in hierarchical systems. The spaces here can interact with each other, but no mechanism has been put forth that allows for a parent-child relationship between systems, concepts, or problems. While not directly applicable, the complex design spaces discussed in Bowen and Dittmar (2017) offer an interesting template that might be altered to work with the spaces as defined here.

Also, several methods have been mentioned for exploring spaces which tend to be based in some form of guided sampling of the space. Even the genetic algorithms of co-evolution are a type of sampling. However, there are methods in other arenas for estimating the pre-image of multivariate functions that may be useful here. The first known information in design is generally in regard to the problem, yet the problem map *outputs* to the problem space. So having a mechanism for reversing the relationship could be highly desirable.

Another issue is the 'curse of dimensionality' as it is often called in the field of machine learning. Figure 3.14 depicts 78 separate pairwise plots, yet the example in question had only seven requirements and 13 design variables. In practice, requirements often number in the hundreds or thousands and design variables may well enter the millions. While it is not strictly necessary to

visualize the proposed spaces in their entirety to conduct an effective analysis, sampling a useful quantity of points in the form space may be intractable without access to high-performance computing facilities. And even simple tasks like choosing a search space becomes daunting. A more reasonable approach might be to employ dimensional reduction to offset some of this burden. The advent of modern machine learning techniques and tools has yielded a variety of approaches to this problem which can be adapted for use here as well.

Finally, another discipline from which useful techniques might be borrowed is that of topological data analysis (TDA). Like machine learning, this field has also experienced tremendous growth in recent years and many of its techniques operate on topological spaces similar to those used in this paper. Data scientists use TDA to analyze, understand, and find patterns in extremely large quantities of data. In the abstract, that is essentially what the points in these design spaces constitute, so it stands to reason that at least some of the tools in that area can be applied here with the right adaptation and interpretation.

Bibliography

Anandan, S., Teegavarapu, S., and Summers, J. D. (2006). Issues of similarity in engineering design. In *International design engineering technical conferences and computers and information in engineering conference*, volume 4255, pages 73–82.

ANSYS, Inc. (2020). Ansys mechanical 2020 r2.

Blonder, B. (2018). Hypervolume concepts in niche-and trait-based ecology. *Ecography*, 41(9):1441–1455.

Borggaard, J. T. (1994). *The sensitivity equation method for optimal design*. PhD thesis, Virginia Tech.

Bowen, J. and Dittmar, A. (2017). Formal definitions for design spaces and traces. In *2017 24th Asia-Pacific Software Engineering Conference (APSEC)*, pages 600–605. IEEE.

Braha, D. and Reich, Y. (2003). Topological structures for modeling engineering design processes. *Research in Engineering Design*, 14:185–199.

Broder, A. Z. (1997). On the resemblance and containment of documents. In *Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No. 97TB100171)*, pages 21–29. IEEE.

Bucciarelli, L. L. (2009). *Engineering mechanics for structures*. Courier Dover Publications.

Carson, R. S. (2015). Implementing structured requirements to improve requirements quality. In *INCOSE International Symposium*, volume 25, pages 54–67. Wiley Online Library.

Croom, F. H. (2016). *Principles of topology*. Courier Dover Publications.

Dinar, M., Shah, J., Hunt, G., Campana, E., and Langley, P. (2011). Towards a formal representation model of problem formulation in design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 54860, pages 263–272.

Gero, J. S. and Kannengiesser, U. (2004). The situated function–behaviour–structure framework. *Design studies*, 25(4):373–391.

Goel, V. and Pirolli, P. (1992). The structure of design problem spaces. *Cognitive science*, 16(3):395–429.

Graff, L., Harbrecht, H., and Zimmermann, M. (2016). On the computation of solution spaces in high dimensions. *Structural and Multidisciplinary Optimization*, 54(4):811–829.

Hatchuel, A. and Weil, B. (2009). Ck design theory: an advanced formulation. *Research in engineering design*, 19(4):181–192.

Hijmans, R. J., Cameron, S. E., Parra, J. L., Jones, P. G., and Jarvis, A. (2005). Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 25(15):1965–1978.

Hirshorn, S. R., Voss, L. D., and Bromley, L. K. (2017). Nasa systems engineering handbook. Technical Report NASA/SP-2016-6105 REV 2, NASA.

INCOSE (2015). Guide for writing requirements. *Version 2. Prepared by: Requirements Working Group*.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50.

MacLean, A., Young, R. M., Bellotti, V. M., and Moran, T. P. (1991). Questions, options, and criteria: Elements of design space analysis. *Human–computer interaction*, 6(3-4):201–250.

Maher, M. L., Poon, J., and Boulanger, S. (1996). Formalising design exploration as co-evolution. In *Advances in formal design methods for CAD*, pages 3–30. Springer.

MATLAB (2020). *version 9.8.0 (R2020a)*. The MathWorks Inc., Natick, Massachusetts.

Miller, S. W., Simpson, T. W., Yukish, M. A., Stump, G., Mesmer, B. L., Tibor, E. B., Bloebaum, C. L., and Winer, E. H. (2014). Toward a value-driven design approach for complex engineered systems using trade space exploration tools. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 46315, page V02AT03A052. American Society of Mechanical Engineers.

Mott, R. L. (1999). *Machine Elements in Mechanical Design*. Prentice Hall, Upper Saddle River, N.J, third edition.

Ortiz, J., Summers, J., Coykendall, J., Roberts, T., and Rai, R. (2021). A Topological Formalism for Quantitative Analysis of Design Spaces. (in press).

Pahl, G. and Beitz, W. (2013). *Engineering design: a systematic approach*. Springer Science & Business Media.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Roberts, T. (2021). Design of a polymer fdm testbed for in situ part monitoring via photogrammetry. Master's thesis, Clemson University. (research in progress).

Rosen, D. and Peters, T. (1996). The role of topology in engineering design research. *Research in Engineering Design*, 8(2):81–98.

Ruiz-Pérez, L., Messager, L., Gaitzsch, J., Joseph, A., Sutto, L., Gervasio, F. L., and Battaglia, G. (2016). Molecular engineering of polymersome surface topology. *Science advances*, 2(4):e1500948.

Schätz, B., Hölzl, F., and Lundkvist, T. (2010). Design-space exploration through constraint-based model-transformation. In *2010 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems*, pages 173–182. IEEE.

Shan, S. and Wang, G. G. (2004). Space exploration and global optimization for computationally intensive design problems: a rough set based approach. *Structural and Multidisciplinary Optimization*, 28(6):427–441.

Siddiqi, A., Bounova, G., de Weck, O. L., Keller, R., and Robinson, B. (2011). A posteriori design change analysis for complex engineering projects. *Journal of Mechanical Design*, 133(10).

Siddique, Z. and Rosen, D. W. (2001). On combinatorial design spaces for the configuration design of product families. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AI EDAM*, 15(2):91.

Simpson, G. G. (1943). Mammals and the nature of continents. *American Journal of Science*, 241(1):1–31.

Snášel, V., Nowaková, J., Xhafa, F., and Barolli, L. (2017). Geometrical and topological approaches to big data. *Future Generation Computer Systems*, 67:286–296.

Suh, N. P. (1998). Axiomatic design theory for systems. *Research in engineering design*, 10(4):189–209.

Summers, J. D. (2005). Reasoning in engineering design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 4742, pages 329–340.

Tan, J. J., Otto, K. N., and Wood, K. L. (2017). Relative impact of early versus late design decisions in systems development. *Design Science*, 3.

Taura, T. and Yoshikawa, H. (1994). Managing function concepts in the design process. In *Management of Design*, pages 179–203. Springer.

Wagner, H. and Dłotko, P. (2014). Towards topological analysis of high-dimensional feature spaces. *Computer Vision and Image Understanding*, 121:21–26.

Wang, R., Nellippallil, A. B., Wang, G., Yan, Y., Allen, J. K., and Mistree, F. (2018). Systematic design space exploration using a template-based ontological method. *Advanced Engineering Informatics*, 36:163–177.

Wasserman, L. (2018). Topological data analysis. *Annual Review of Statistics and Its Application*, 5:501–532.

Weisstein, E. W. Point-set topology. Wolfram MathWorld. `https://mathworld.wolfram.com/Point-SetTopology.html`. Accessed: 2021-06-11.

Zhu, J. and Gao, T. (2016). *Topology optimization in engineering structure design*. Elsevier.

Zimmermann, M. and von Hoessle, J. E. (2013). Computing solution spaces for robust design. *International Journal for Numerical Methods in Engineering*, 94(3):290–307.